**Classification of X-Ray Images for Pneumonia and COVID-19 using Deep Learning Models**

Final Year Project Report by

**Asad Ali (023-20-0115)**

**Muhammad Hussain (023-20-0076)**

**Um-E-Hani (023-20-0079)**

Project Code 20F-23

Supervisor: Dr. Ghulam Murtaza Memon

In Partial Fulfillment of the Requirements for the degree Bachelor of Science (CS/SE)

Sukkur IBA University Sukkur, Sindh

2024

**Classification of X-Ray Images for Pneumonia and COVID-19 using Deep Learning Models**

Submitted to the Department of Computer Science in Partial Fulfillment of the Requirements for the Degree of

**Bachelor of Science in Computer Science/ Software Engineering**

At

**Sukkur IBA University, Sukkur, Sindh**

May 2024

**© Asad Ali, Muhammad Hussain, Um-E-Hani**. All rights reserved.

The author hereby grants permission to Sukkur IBA to reproduce and distribute publicly paper and electronic copies of this thesis and grant others the right to do so.

Signature of Authors

Certified by: Internal Examiner

Dr.Zakria, Designation, Department of Computer Science/ Software Engineering

Dr.Faisal, Designation, Department of Computer Science/ Software Engineering

External Examiner

Accepted by

**DECLARATION**

We hereby declare that this project report entitled “Classification of X-ray Images for Pneumonia and COVID-19 using Deep Learning Models” submitted to the Department of Computer Science/ Software Engineering is the record of an original work done by as under the guidance of Dr. Ghulam Murtaza Memon and that no part has been plagiarized without citations. Also, this project is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Computer Science.

**Team Members** **Signature**

Asad Ali

Muhammad Hussain

Um-E-Hani

**Supervisor** **Signature**

Dr.Ghulam Murtaza Memon

Date: /05/2024

**Dedication**

We dedicate this final year project to our parents, who have been our constant source of support and inspiration during the academic journey with their steadfast love and prayers. Their affection and support have served as the cornerstones around which our dreams have been constructed. We also want to express our sincere gratitude to all our wonderful teachers, especially our dedicated supervisor, whose invaluable guidance and unwavering commitment were crucial in ensuring the success of this project. At every turn, their guidance, insight, and support have been invaluable, and we are incredibly appreciative of all that they have done to support our intellectual and personal development. This accomplishment is a result of the combined efforts of our parents and instructors, to whom we sincerely thank and show respect.

**Acknowledgements**

We would like to express our sincere appreciation to Dr Ghulam Murtaza Memon, our supervisor, for his constant leadership and insightful advice during the project's progress. Their guidance has greatly influenced our work, and we are sincerely appreciative of their knowledge.

We are truly grateful to our parents, whose love and prayers have been our unwavering support. Your confidence in our talents has motivated us, and we appreciate your support in helping us accomplish this academic goal. We also thank the CRAIB lab for their support, which was crucial to this project's successful conclusion.

This accomplishment is the result of teamwork, and we are grateful to friends, classmates, and the whole Sukkur IBA Computer Science community for all of their support, encouragement, and friendship. Every individual who has contributed to our academic path is deeply appreciated for their part, since they have all been instrumental in our accomplishment.

Table of Contents

[1. Introduction 11](#_Toc162822836)

[2. Project and Research Objectives 13](#_Toc162822837)

[2.1. Project Objectives 13](#_Toc162822838)

[2.2. Research Objectives 13](#_Toc162822839)

[3. Literature Review 14](#_Toc162822840)

[3.1. COVID-Net: 14](#_Toc162822841)

[3.2. DenseNet: 15](#_Toc162822842)

[3.3. ResNet-50: 15](#_Toc162822843)

[4. Problem Definition 16](#_Toc162822844)

[*5.* Methodology/Solution 17](#_Toc162822845)

[5.1. Data Preprocessing: 17](#_Toc162822846)

[5.2. Data Splitting: 18](#_Toc162822847)

[5.3. Model Tuning: 18](#_Toc162822848)

[5.4. Training Model and Evaluating Model: 18](#_Toc162822849)

[5.4.1. Version 0: 19](#_Toc162822850)

[5.4.2. Version 1: 19](#_Toc162822851)

[5.4.3. Version 2: 20](#_Toc162822852)

[5.4.4. Version 3: 20](#_Toc162822853)

[5.4.5. Version 4: 21](#_Toc162822854)

[5.4.6. Version 5: 21](#_Toc162822855)

[5.4.7. Version 6: 22](#_Toc162822856)

[5.4.8. Version 7: 22](#_Toc162822857)

[5.5. Web Application-Front End: 23](#_Toc162822858)

[5.6. API Creation and Back End: 23](#_Toc162822859)

[5.7. Integrating Model: 23](#_Toc162822860)

[5.8. Deployment: 23](#_Toc162822861)

[6. Detailed Design and Architecture 24](#_Toc162822862)

[6.1. System Architecture: 24](#_Toc162822863)

[6.1.1. Classification Model for Quality: 24](#_Toc162822864)

[6.1.2. VGG16: 24](#_Toc162822865)

[6.1.3. ResNet50: 25](#_Toc162822866)

[6.1.4. MobileNet: 25](#_Toc162822867)

[6.2. System Constraints: 25](#_Toc162822868)

[6.2.1. Software Constraints: 25](#_Toc162822869)

[6.2.2. Hardware Constraints: 25](#_Toc162822870)

[6.2.3. Cultural Constraints: 26](#_Toc162822871)

[6.3. Flow Diagram: 26](#_Toc162822872)

[6.4. Use Case Diagram: 26](#_Toc162822873)

[6.5. Sequence Diagram: 27](#_Toc162822874)

[6.6. Entity Relationship Diagram: 28](#_Toc162822875)

[6.7. Interface Design: 29](#_Toc162822876)

[7. Implementation and Testing 31](#_Toc162822877)

[7.1. Version 0: 32](#_Toc162822878)

[7.2. Version 1: 34](#_Toc162822879)

[7.3. Version 2: 35](#_Toc162822880)

[7.4. Version 3: 36](#_Toc162822881)

[7.5. Version 4: 37](#_Toc162822882)

[7.6. Version 5: 38](#_Toc162822883)

[7.7. Version 6: 39](#_Toc162822884)

[7.8. Version 7: 40](#_Toc162822885)

[8. Results and Discussion 41](#_Toc162822886)

[8.1. Findings and Problems Version 0: 41](#_Toc162822887)

[8.2. Findings and Problems Version 1: 42](#_Toc162822888)

[8.3. Findings & Problems Version 2: 42](#_Toc162822889)

[8.4. Findings & Problems Version 3: 43](#_Toc162822890)

[8.5. Findings & Problems Version 4: 45](#_Toc162822891)

[8.6. Findings & Problems Version 5: 47](#_Toc162822892)

[8.7. Findings & Problems Version 6: 49](#_Toc162822893)

[8.8. Findings & Problems Version 7: 51](#_Toc162822894)

[9. Project Timeline 54](#_Toc162822895)

[10. Conclusion and Future Work 55](#_Toc162822896)

[11. References 56](#_Toc162822897)

**List of Figures**

[Figure 1: Methodology 18](#_Toc162823030)

[Figure 2: Data Augmentation Techniques 19](#_Toc162823031)

[Figure 3: System Architecture Diagram of VGG-16 25](#_Toc162823032)

[Figure 4: Architecture Diagram of ResNet50 26](#_Toc162823033)

[Figure 5: Architecture Diagram of MobileNetV3 26](#_Toc162823034)

[Figure 6: Flow Diagram 27](#_Toc162823035)

[Figure 7: Use Case Diagram 28](#_Toc162823036)

[Figure 8: Sequence Diagram 29](#_Toc162823037)

[Figure 9: Entity Relationship Diagram 30](#_Toc162823038)

[Figure 10: About Us Page 31](#_Toc162823039)

[Figure 11: Home Page 31](#_Toc162823040)

[Figure 12: Records Page 31](#_Toc162823041)

[Figure 13: Home Page Mobile View 31](#_Toc162823042)

[Figure 14: Sign Up Page 32](#_Toc162823043)

[Figure 15: Login Page 32](#_Toc162823044)

[Figure 16: Confusion Matrix Version 0 33](#_Toc162823045)

[Figure 17: Classification Report Version 0 34](#_Toc162823046)

[Figure 18: Confusion Matrix Version 1 35](#_Toc162823047)

[Figure 19: Classification Report Version 1 36](#_Toc162823048)

[Figure 20: Confusion Matrix Version 2 36](#_Toc162823049)

[Figure 21: Classification Report Version 2 37](#_Toc162823050)

[Figure 22: Confusion Matrices for Folds 1,2,3 of Version 3 37](#_Toc162823051)

[Figure 23: Confusion Matrices for Fold 4,5 of Version 3 38](#_Toc162823052)

[Figure 24: Confusion Matrix for Folds 1,2,3 for Version 4 38](#_Toc162823053)

[Figure 25: Confusion Matrices for Folds 4,5 for Version 4 39](#_Toc162823054)

[Figure 26: Confusion Matrices for Folds 1,2,3 for Version 5 39](#_Toc162823055)

[Figure 27: Confusion Matrices for Folds 4,5 for Version 5 40](#_Toc162823056)

[Figure 28:Confusion Matrices for Folds 1,2,3 for Version 6 40](#_Toc162823057)

[Figure 29: Confusion Matrices for Folds 4,5 for Version 6 41](#_Toc162823058)

[Figure 30: Confusion Matrices for Folds 1,2,3 for Version 7 41](#_Toc162823059)

[Figure 31: Confusion Matrices for Folds 4,5 for Version 7 42](#_Toc162823060)

[Figure 32: Accuracy & Loss Graphs Version 0 42](#_Toc162823061)

[Figure 33: Accuracy & Loss Graphs Version 1 43](#_Toc162823062)

[Figure 34: Accuracy & Loss Graphs Version 2 43](#_Toc162823063)

[Figure 35: Accuracy & Loss fold 1 Version 3 44](#_Toc162823064)

[Figure 36: Accuracy & Loss fold 2 Version 3 44](#_Toc162823065)

[Figure 37: Accuracy & Loss fold 4 Version 3 45](#_Toc162823066)

[Figure 38: Accuracy & Loss fold 3 Version 3 45](#_Toc162823067)

[Figure 39: Accuracy & Loss fold 5 Version 3 46](#_Toc162823068)

[Figure 40: Accuracy & Loss fold 1 Version 4 46](#_Toc162823069)

[Figure 41: Accuracy & Loss fold 2 Version 4 47](#_Toc162823070)

[Figure 42: Accuracy & Loss fold 3 Version 4 47](#_Toc162823071)

[Figure 43: Accuracy & Loss fold 4 Version 4 47](#_Toc162823072)

[Figure 44: Accuracy & Loss fold 5 Version 4 48](#_Toc162823073)

[Figure 45: Accuracy & Loss fold 1 Version 5 48](#_Toc162823074)

[Figure 46: Accuracy & Loss fold 2 Version 5 49](#_Toc162823075)

[Figure 47: Accuracy & Loss fold 3 Version 5 49](#_Toc162823076)

[Figure 48: Accuracy & Loss fold 4 Version 5 49](#_Toc162823077)

[Figure 49: Accuracy & Loss fold 5 Version 5 50](#_Toc162823078)

[Figure 50: Accuracy & Loss for Fold 1 Version 6 50](#_Toc162823079)

[Figure 51: Accuracy & Loss for Fold 2 Version 6 51](#_Toc162823080)

[Figure 52: Accuracy & Loss for Fold 3 Version 6 51](#_Toc162823081)

[Figure 53: Accuracy & Loss for Fold 4 Version 6 51](#_Toc162823082)

[Figure 54: Accuracy & Loss for Fold 5 Version 6 52](#_Toc162823083)

[Figure 55: Accuracy & Loss fold 1 Version 7 52](#_Toc162823084)

[Figure 56: Accuracy & Loss fold 2 Version 7 53](#_Toc162823085)

[Figure 57: Accuracy & Loss fold 3 Version 7 53](#_Toc162823086)

[Figure 58: Accuracy & Loss fold 4 Version 7 53](#_Toc162823087)

[Figure 59: Accuracy & Loss fold 5 Version 7 54](#_Toc162823088)

**List of tables**

[Table 1: Version 0 16](#_Toc162700085)

[Table 2: Version 1 16](#_Toc162700086)

[Table 3: Version 2 17](#_Toc162700087)

[Table 4: Version 3 17](#_Toc162700088)

[Table 5: Version 4 17](#_Toc162700089)

[Table 6: Version 5 18](#_Toc162700090)

[Table 7: Version 6 18](#_Toc162700091)

**Abstract**

Respiratory diseases, particularly pneumonia and COVID-19, continue to pose significant public health challenges worldwide. Rapid and accurate diagnosis is crucial for effective patient management and disease control. This project aims to develop and evaluate deep learning models for the automated classification of chest X-ray images into three categories: pneumonia, COVID-19, and normal cases. The publicly available "COVID-19+PNEUMONIA+NORMAL Chest X-Ray Image Dataset" from Kaggle will be utilized for training and testing the models([Sachin Kumar, 2022](#_ENREF_10); [Shastri, 2022](#_ENREF_11)).

In addition to model development, we will create a user-friendly web interface that allows physician to upload chest X-ray images. This interface will enable easy access to our diagnostic tool, making it accessible to healthcare professionals and the public.

Convolutional Neural Networks (CNNs), data augmentation, transfer learning, and attention mechanisms will be employed to enhance model performance and interpretability. The models will undergo rigorous evaluation using various performance metrics.

This project has the potential to contribute significantly to X-ray image analysis by providing a reliable and accessible tool for the early detection and differentiation of pneumonia and COVID-19 from chest X-ray images.

# Introduction

As of UNICEF’s report of 2019([United Nations International Children's Emergency Fund, 2019](#_ENREF_12)) , Estimated number of pneumonia-related deaths in children under five are 58,000 making Pakistan the number third in the list after Nigeria and India. The Express Tribune claims that 92,000 children die from pneumonia every year in Pakistan, and it accounts for approximately 20% of childhood deaths worldwide. Though Pneumonia is the most common chest/lung’s disease the Covid-19; also, a chest/lung’s disease, caused more than six million deaths worldwide([World Health Organization, 2020](#_ENREF_14)) and 30,656 deaths in Pakistan([Organization, 2023](#_ENREF_8)). However, it is assumed that deaths could be lesser if the disease could have been detected earlier.

In the sphere of medical sciences, the ability to identify diseases from X-ray images has emerged as a promising tool. But classifying chest X-ray images—differentiating between pneumonia, Covid-19, and other illnesses is a difficult task. as it requires domain knowledge in addition to expertise, like what a highly skilled and experienced radiologist would have. Nevertheless, the emergence of artificial intelligence (AI) has completely changed the field of health sciences, especially regarding the use of X-ray picture classification for Covid-19 and pneumonia. A doctor can now prescribe a radiological examination (X-ray) instead of needing to schedule several consultations, which streamlines the diagnosis procedure by enabling an automatically classified model trained with Deep Learning to identify images. This relieves radiologists of the laborious work of accurately analyzing and classifying X-ray pictures, as well as the time constraints faced by doctors. With significant consequences for public health, the use of AI to medical diagnostics holds great promise for improving the effectiveness and precision of illness detection.

A cornerstone of radiological imaging for many decades, chest radiography (chest X-ray, CXR) remains the most commonly performed radiological exam in the world with industrialized countries reporting an average 238 erect-view chest X-ray images acquired per 1000 of population annually (United Nations, 2008)([Erdi Çallı 2021](#_ENREF_5)). This huge number shows the importance of X-ray Images in the field of health sciences.

Radiology involves analyzing medical images such as X-rays, MRIs, or CT scans to diagnose diseases or injuries. However, this process is not always straightforward and can be subject to uncertainty([Adrian Brady, 2012](#_ENREF_2)). Radiologists must interpret images based on the information available, which may sometimes be limited or unclear. Additionally, medical imaging can present challenges such as overlapping structures, image artifacts, or variations in patient positioning, which can further contribute to uncertainty in interpretation. Therefore, despite their expertise, radiologists may encounter cases where definitive conclusions are difficult to reach, leading to the recognition that radiology interpretations are not always perfect or infallible.

Delayed presentation of COVID-19 pneumonia increases the risk of mortality and need for high-intensity healthcare. Conversely, early identification of COVID-19 pneumonia grants an opportunity to intervene early and thus prevent more complicated, protracted and less successful hospital admissions([Daniel Goyal1, 2021](#_ENREF_3)).

The need for a system incorporating Deep Learning trained models becomes apparent in addressing the complexities of timely and accurate disease diagnosis. Traditional diagnostic methods often face challenges in handling the increasing volume of medical data and the demand for swift identification of diseases such as pneumonia and Covid-19. The proposed system aims to leverage the power of Deep Learning to automate and enhance the classification process of X-ray images. Deep Learning models, trained on extensive datasets, possess the capacity to discern intricate patterns and variations, enabling them to differentiate between various chest/lung conditions with a high level of accuracy. By implementing such a system, we aspire to not only expedite the diagnostic process but also contribute to the early detection of diseases, thereby mitigating the severity and impact of pneumonia and Covid-19 on public health. The intersection of advanced technology and healthcare, in this context, signifies a significant step toward more efficient, reliable, and accessible diagnostic solutions.

# Project and Research Objectives

## Project Objectives

Develop a user-friendly, interactive AI web-based system that meets the following criteria:

* Features an intuitive interface for Pulmonologists and patients to diagnose Pneumonia and COVID-19.
* Examines the X-ray image and classifies it into any of the three categories i.e. Pneumonia, COVID-10 or normal.

## Research Objectives

To do experimental research that involves:

* Comprehensive study of object classification techniques and its literature.
* An empirical evaluation of various state-of-the-art object classification models, such as the VGG.

# Literature Review

Classification of X-ray images is a crucial aspect in the field of deep learning and computer vision techniques have played a significant role in advancing research in this area. In the landscape of COVID-19 diagnosis, chest X-ray imaging has emerged as a prominent method, attracting considerable research interest. The application of Deep Learning-particularly Convolutional Neural Networks (CNNs), known for their prowess in extracting features automatically from unstructured data like images, has been pivotal in this context. This section provides a comprehensive overview of the existing studies in the field and highlights the key findings.

Corona virus is considered a syndrome of the respiratory system, the outbreak in Wuhan, Hubei Province, China. This virus lately called Coronavirus Disease (COVID-19), recognized world-widely as Global Communal Health Extremity by World Health Organization (WHO) in January 2020, also acquired the status of “Pandemic” on March 11, 2020.   
Pneumonia, traditionally recognized as an inflammatory condition of the lung affecting primarily the microscopic air sacs known as alveoli, was a significant subject of medical research long before the advent of COVID-19. Despite improvements in medical care, pneumonia - a major cause of death globally, mainly affects children and the elderly. The WHO has classified it as a serious public health concern that calls for better methods of diagnosis and treatment.

In the biomedical sciences, artificial intelligence (AI)-based deep learning models have become a key innovation, building on the pressing need for sophisticated diagnostic techniques to battle pneumonia. The capacity of these models to identify a broad variety of illnesses has drawn a lot of interest, demonstrating the potential to completely transform the diagnosis and treatment of illnesses like pneumonia and COVID-19. Recent advances in computer vision, more specifically, image classification have shown promising results in medical settings. Computer vision algorithms are accurate, efficient, and more importantly extremely quick to produce results compared to any radiologists or other imaging modalities currently used in practice. For example, Radiology imaging delays have been found to independently contribute to longer hospital stays, as indicated by recent study([S. Cournane a, 2016](#_ENREF_9)). In addition a separate study found that creating reports from X-rays in non-emergency cases took over hours([Nayana Ambardekar, 2022](#_ENREF_7)).

In the context of using deep learning for the classification of COVID-19 and other respiratory diseases from chest X-ray images, several innovative techniques and architectures have been developed. Few of these are following,

## COVID-Net:

A tailored deep convolutional neural network design specifically created for detecting COVID-19 cases from chest X-ray images. COVID-Net is known for its high accuracy and efficiency, utilizing a novel architecture that prioritizes critical features for COVID-19 detection. It can be observed in a study([Linda Wang, 2020](#_ENREF_6)) that COVID-Net has achieved an impressive accuracy of 93.3%. The dataset used in the study comprised of a total of 13,975 CXR images across 13,870 patient cases. This demonstrates the model's exceptional ability to leverage deep learning for significant advancements in medical diagnostics, particularly in the rapid and accurate identification of COVID-19, thereby contributing to effective patient management and control of the pandemic spread.

## DenseNet:

DenseNet (Densely Connected Convolutional Network) improves upon the idea of feature reuse, making each layer directly connected to every other layer in a feed-forward fashion. For medical imaging, this architecture allows for substantial reductions in parameters and improvements in efficiency, potentially enhancing the model's performance on chest X-ray datasets as found in a study([Elene Firmeza Ohata, Bessa Albuquerque, & Pedro Pedrosa Rebouças Filho, 2021](#_ENREF_4)) using DenseNet with Transfer Learning achieved an accuracy of 98.205% and minimum F1-score of 98.205% . Author here, made two types of data splits:

according to datasets: A (with augmentation) and B,

according to classes: COVID-19 images and healthy images.

Images in each dataset were 388.

## ResNet-50:

Resnet (Residual Neural Network) With its deep residual learning framework, ResNet can be effectively used for image classification tasks, including medical image analysis. Its architecture facilitates the training of much deeper networks by using skip connections or shortcuts to jump over some layers. Up to 96.2% accuracy has been achieved by using ResNet and Recall of 98.4%([A. Victor Ikechukwu, 2021](#_ENREF_1)).

The decision to undertake this project is rooted in a deep recognition of the urgent need for advancements in disease diagnosis, particularly in the context of pneumonia and Covid-19 detection, as mentioned earlier. With pneumonia being a leading cause of childhood mortality and Covid-19 causing a global health crisis, the statistics presented highlight the need for more efficient and accurate diagnostic techniques. It is further illuminated that the transformative potential of computer vision techniques, particularly Deep Learning models, in addressing the complexities of disease diagnosis from chest X-ray images. By leveraging these advancements, our project seeks to develop a system that automates and enhances the classification process of X-ray images, aiming to expedite diagnosis, facilitate early disease detection, and ultimately mitigate the severity and impact of pneumonia and Covid-19 on public health. Through our dedication to use advanced technology for healthcare, we strive to contribute to improved health outcomes and the advancement of medical diagnostics.

# Problem Definition

In the areas with limited resources, particularly where access to skilled medical professionals is scarce, the swift and accurate diagnosis of respiratory illnesses such as Pneumonia and Covid-19 presents a formidable challenge. While radiological images, notably X-rays, are crucial for early disease detection and monitoring, the shortage of trained radiologists results in diagnostic delays, posing potential risks to patients. The conventional manual approach to image interpretation and disease diagnosis is both time-consuming and vulnerable to human error, particularly under the strain of high patient-to-physician ratios. Given the critical nature of respiratory diseases and the profound impact of timely diagnosis on patient outcomes, there exists an urgent need for automated solutions. Thus, our project endeavors to develop a user-friendly system leveraging Convolutional Neural Networks (CNNs) to categorize Chest X-ray Images as either Pneumonia-affected, Covid-19-affected, or devoid of these conditions. By reducing reliance on expert radiologists and streamlining respiratory disease detection, our system aims to bridge gaps in healthcare accessibility and improve diagnostic efficiency in resource-limited settings.

# Methodology/Solution



Figure 1: Methodology

## Data Preprocessing:

Data preprocessing plays a crucial role in enhancing the quality and diversity of the dataset, thereby improving the robustness and generalization capabilities of the model. In our project, we employed data augmentation techniques as part of the preprocessing pipeline to augment the training data. By applying various augmentation techniques such as rescaling, rotation, width shift, height shift, shearing, and flipping, we aimed to introduce variability and diversity into the dataset. Rescaling ensures that the pixel values of the images are within a certain range, facilitating consistency and normalization across the dataset. Rotation allows for the introduction of different orientations of the images, mimicking real-world scenarios where patients may not be perfectly aligned during imaging. Similarly, width and height shift techniques enable the translation of the image along the width and height dimensions, introducing spatial variability. Shearing introduces affine transformations to the images, modifying their shapes and perspectives. Flipping horizontally or vertically adds mirror images to the dataset, further enriching its diversity. By incorporating these augmentation techniques, we sought to prevent overfitting and improve the model's ability to generalize to unseen data, ultimately contributing to more robust and reliable classification results. Following is a referenced image for augmentation techniques([Varde, 2023](#_ENREF_13)).

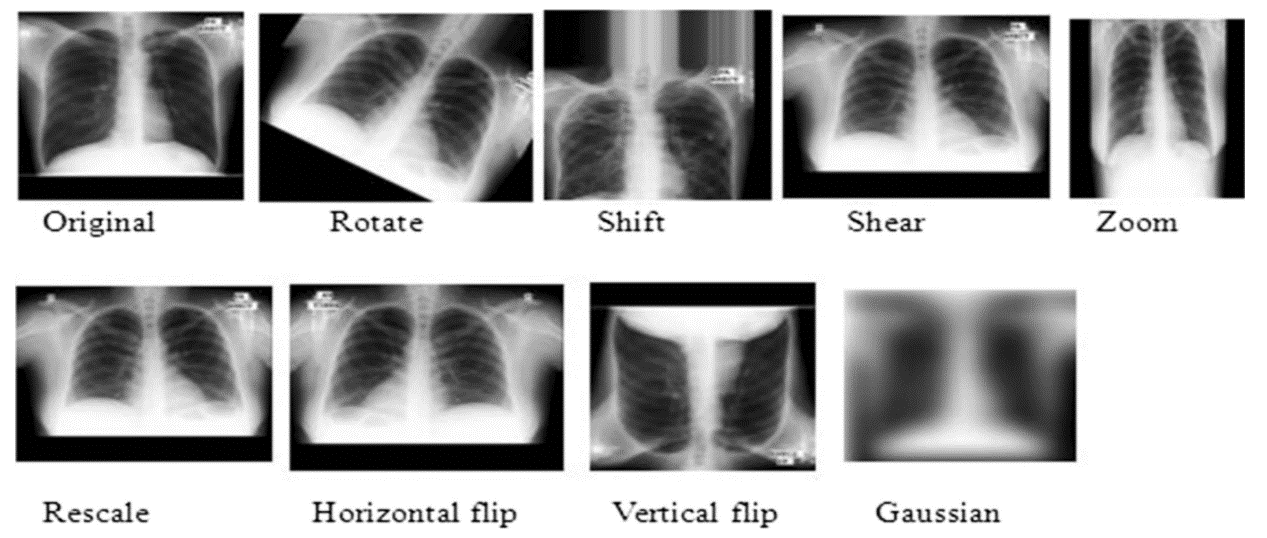


Figure 2: Data Augmentation Techniques

## **Data Splitting:**

Data splitting to perform 5-fold cross-validation is a crucial step in evaluating the performance of machine learning models. In this process, the available dataset is divided into five equally sized subsets, or folds. Each fold serves as a separate validation set while the remaining four folds are used for training the model. This procedure is repeated five times, with each fold used once as the validation set. By systematically rotating through each fold as the validation set, the model is trained and evaluated on different subsets of the data, providing a robust assessment of its performance. This approach helps to mitigate the risk of overfitting and provides a more reliable estimate of the model's generalization ability. Additionally, it ensures that the model's performance is not overly dependent on the specific partitioning of the data, thereby enhancing its reliability and applicability to new datasets.

## Model Tuning:

Model tuning is an essential aspect of machine learning model development, aimed at optimizing the model's performance and generalization ability. During model tuning, various hyperparameters and settings are adjusted to achieve the best possible results. This process involves systematically exploring different combinations of hyperparameters, such as learning rate, regularization strength, and network architecture. The choice of optimizer plays a significant role in the training process of machine learning models, influencing their convergence speed and overall performance. In our project, we utilized two commonly employed optimizers: Adam and Stochastic Gradient Descent (SGD). Both optimizers offer distinct advantages, with Adam being known for its adaptive learning rates and momentum, while SGD is more traditional and computationally efficient. Additionally, we maintained a consistent learning rate of 0.001 throughout the training process to control the magnitude of parameter updates. Furthermore, to balance computational efficiency and model stability, we employed a batch size of 32, determining the number of samples processed before updating the model's parameters.

## Training Model and Evaluating Model:

The model went through multiple rounds of training, each with its own unique

approach and focus. We meticulously documented three versions of this training process, providing all the necessary details to ensure thorough understanding and transparency. These iterations symbolize our dedication to perfecting the model, as we continuously fine-tuned and adjusted its parameters for optimal performance. We used three different models in our system i.e. ResNet50, MobileNetV3 and VGG-16. With each round of training, we gained valuable insights and made improvements to enhance the model's accuracy and effectiveness.

In our training process, we employed the concept of "Transfer Learning," a powerful technique in machine learning where knowledge gained from training one model is transferred and applied to a different, but related, task. This approach allowed us to use the pre-trained VGG16 model, which was initially developed for general image recognition tasks, and fine-tune it for our specific task of classifying chest X-ray images for pneumonia and COVID-19 detection. By building upon the existing knowledge embedded in the VGG16 model, we were able to accelerate our training process and achieve superior performance with less data. This strategic use of Transfer Learning not only optimized our training efforts but also enhanced the accuracy and efficiency of our model, ultimately contributing to more reliable and timely diagnoses in healthcare settings. Since we trained our model multiple times also we used different pretrained models, the notable processing and findings are following.

### Version 0:

In this version, we employed the ResNet-50 model and introduced an extra Dense layer with 512 nodes. The parameters included Stochastic Gradient Descent optimizer, a learning rate of 0.001, and a batch size of 32. Despite data augmentation, the model achieved 77% accuracy over 100 epochs.

|  |  |
| --- | --- |
| Model | ResNet-50 |
| Modification | Add an extra Dense layer with 512 nodes on top of it. |
| Epochs | 100 |
| Optimizer | Stochastic Gradient Descent |
| Learning Rate | 0.001 |
| Momentum | 0.9 |
| Batch Size | 32 |
| Data Distribution | 70% Training, 15% Validation, 15% Testing |
| Data Augmentation | Yes, applied |
| Total Parameters | 24,638,339 |
| Trainable Parameters | 1,050,627 |
| Non-Trainable Parameters | 23,587,712 |
| Test Accuracy | 77% |

Table 1: Version 0

### Version 1:

For this iteration, we switched to the VGG-16 model and added an extra Dense layer with 1024 nodes. The optimization algorithm remained Stochastic Gradient Descent with a learning rate of 0.001 and a momentum of 0.9. Although the model showed an improvement, achieving 82% accuracy, the training duration was reduced to 10 epochs.

|  |  |
| --- | --- |
| Model | VGG-16 |
| Modification | Add an extra Dense layer with 1024 nodes on top of it. |
| Epochs | 10 |
| Optimizer | Stochastic Gradient Descent |
| Learning Rate | 0.001 |
| Momentum | 0.9 |
| Batch Size | 32 |
| Data Distribution | 70% Training, 15% Validation, 15% Testing |
| Data Augmentation | Yes, applied |
| Total Parameters | 15,243,075 |
| Trainable Parameters | 528,387 |
| Non-Trainable Parameters | 14,714,688 |
| Test Accuracy | 82% |

Table 2: Version 1

### Version 2:

Version 2 continued to utilize the VGG-16 model but introduced two extra Dense layers with 512 and 1024 nodes, respectively. The optimizer was changed to Adam, maintaining a learning rate of 0.001 and a batch size of 32. With these modifications, the model achieved a remarkable 95% accuracy over 50 epochs.

|  |  |
| --- | --- |
| Model | VGG-16 |
| Modification | Added two extra Dense layers with 512,1024 nodes respectively. |
| Epochs | 50 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Batch Size | 32 |
| Data Distribution | 70% Training, 15% Validation, 15% Testing |
| Data Augmentation | Yes, applied |
| Total Parameters | 15,505,731 |
| Trainable Parameters | 791,043 |
| Non-Trainable Parameters | 14,714,688 |
| Test Accuracy | 95% |

Table 3: Version 2

### Version 3:

In this version, we again employed the VGG-16 model and retained the modifications from the previous version. However, we implemented 5-fold cross-validation during training to ensure robustness. The accuracy across different folds varied slightly, with the highest accuracy of 95% observed in fold 3.

|  |  |
| --- | --- |
| Model | VGG-16 |
| Modification | Added two extra Dense layers with 512,1024 nodes respectively. |
| Epochs | 50 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Batch Size | 32 |
| Data Distribution | 70% Training, 15% Validation, 15% Testing along with 5-fold cross validation applied |
| Test Accuracy Fold 1 | 92% |
| Test Accuracy Fold 2 | 94% |
| Test Accuracy Fold 3 | 95% |
| Test Accuracy Fold 4 | 93% |
| Test Accuracy Fold 5 | 93% |

Table 4: Version 3

### Version 4:

Version 4 also utilized the VGG-16 model with two extra Dense layers, each with 4096 nodes. A dropout rate of 0.5 was introduced to reduce overfitting. Despite these changes, the model's accuracy ranged between 90% and 93% across different folds.

|  |  |
| --- | --- |
| Model | VGG-16 |
| Modification | Added two extra Dense layers with 4096 nodes in each. layer |
| Drop Out | 0.5 |
| Epochs | 50 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Batch Size | 32 |
| Data Distribution | 70% Training, 15% Validation, 15% Testing along with 5-fold cross validation applied |
| Test Accuracy Fold 1 | 92% |
| Test Accuracy Fold 2 | 91% |
| Test Accuracy Fold 3 | 92% |
| Test Accuracy Fold 4 | 90% |
| Test Accuracy Fold 5 | 93% |

Table 5: Version 4

### Version 5:

For this iteration, we reverted to the ResNet-50 model and added two extra Dense layers with 4096 nodes each. The dropout rate remained at 0.5, and the optimization algorithm was Adam. However, the model's performance was less satisfactory, achieving accuracies between 66% and 68%.

|  |  |
| --- | --- |
| Model | ResNet-50 |
| Modification | Added two extra Dense layers with 4096 nodes in each. layer |
| Drop Out | 0.5 |
| Epochs | 50 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Batch Size | 32 |
| Data Distribution | 70% Training, 15% Validation, 15% Testing along with 5-fold cross validation applied |
| Test Accuracy Fold 1 | 67% |
| Test Accuracy Fold 2 | 66% |
| Test Accuracy Fold 3 | 68% |
| Test Accuracy Fold 4 | 68% |
| Test Accuracy Fold 5 | 67% |

Table 6: Version 5

### Version 6:

In this iteration, MobileNet was employed with consistent learning rate, optimizer, and momentum parameters. The peak accuracy of 57% was observed during fold 4 of the cross-validation process, showcasing its performance within the dataset.

|  |  |
| --- | --- |
| Model | Custom |
| Epochs | 100 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Momentum | 0.9 |
| Batch Size | 32 |
| Data Distribution | 70% Training, 15% Validation, 15% Testing along with 5-fold cross validation applied |
| Test Accuracy Fold 1 | 34% |
| Test Accuracy Fold 2 | 34% |
| Test Accuracy Fold 3 | 34% |
| Test Accuracy Fold 4 | 57% |
| Test Accuracy Fold 5 | 34% |

Table 7: Version 6

### Version 7:

In the final version, a custom model architecture was employed, featuring various Conv2D and Dense layers. The model achieved high accuracy, ranging from 92% to 96% across different folds, demonstrating the effectiveness of the custom design.

|  |  |
| --- | --- |
| Model | Custom |
| Epochs | 100 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Momentum | 0.9 |
| Batch Size | 32 |
| Data Distribution | 70% Training, 15% Validation, 15% Testing along with 5-fold cross validation applied |
| Test Accuracy Fold 1 | 95% |
| Test Accuracy Fold 2 | 93% |
| Test Accuracy Fold 3 | 92% |
| Test Accuracy Fold 4 | 96% |
| Test Accuracy Fold 5 | 94% |

Table 8: Version 7

## Web Application-Front End:

The web application front-end development focused on creating a simple and user-friendly interface. We used modern design techniques to ensure it looks good and works well on different devices. The color themes used align well with the colors used in health sector. Through testing and refinement, we aimed to make it easy for users to navigate and interact with the app.

* 1. API Creation and Back End:

Creating the API and back-end was like building a strong foundation for our project. We made sure it could handle all the behind-the-scenes work, like storing and retrieving data, so the front-end could interact smoothly with it. By using the latest technologies and making sure everything was secure, we aimed to create a reliable system that could handle lots of users without any difficulty. After testing and making improvements, we are confident that our API and back-end are ready to support our project and make it a success.

## Integrating Model:

The trained model is integrated into the Web Application of our X-ray Image Classification into Pneumonia and COVID-19.

## Deployment:

Deploying the model is like putting it into action, making it available for real-world use. We use advanced technology and cloud services to seamlessly integrate the model into our website. Now, doctors can easily use our website to analyze chest X-ray images and quickly identify signs of pneumonia and COVID-19. We've made sure the system can handle lots of users and keep working reliably, no matter where they are. By making our model accessible online, we're helping doctors everywhere make better decisions and detect diseases earlier, which means better care for patients worldwide.

# Detailed Design and Architecture

## System Architecture:

### Classification Model for Quality:

In this process, we employed VGG16, a renowned convolutional neural network architecture, alongside ResNet50 and the concept of Transfer Learning, establishing a robust framework for image classification. In our exploration, various models were considered, yet certain factors steered our selection towards models like VGG16, MobileNetV3 and ResNet50. Models with intricate architectures, such as those with numerous layers, were not chosen due to the limited availability of training data and computational resources like Inception and DenseNet. Training these complex models necessitates a substantial volume of diverse data to capture and learn intricate patterns effectively. However, in our scenario, the dataset size was insufficient to support the training of such large models adequately. Additionally, models with extensive layers demand significant computational power, which was not readily accessible. Therefore, to strike a balance between model complexity, dataset size, and computational resources, we opted for VGG16, ResNet50 and MobileNetV3, leveraging their proven effectiveness in image classification tasks while ensuring feasibility within our constraints.

All of these models leverage Transfer Learning to adapt pre-trained weights, enhancing their ability to classify medical images accurately. Given the complexity of medical images, this multifaceted approach ensures optimal performance and reliability in disease detection and classification tasks.

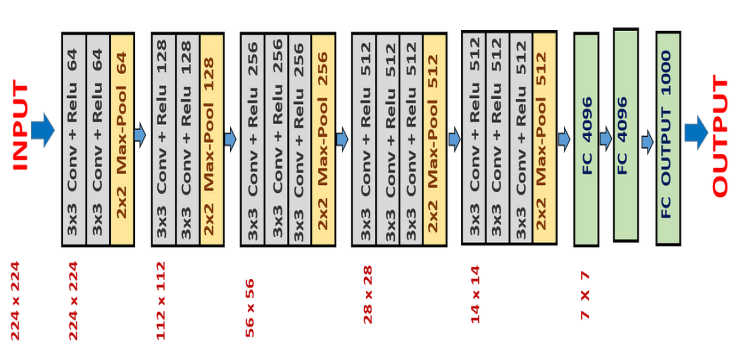


Figure 3: System Architecture Diagram of VGG-16

### VGG16:

VGG16, with its distinctive design featuring 16 layers, including 13 convolutional layers and 3 fully connected layers, stands as a formidable tool in extracting intricate features from input images.

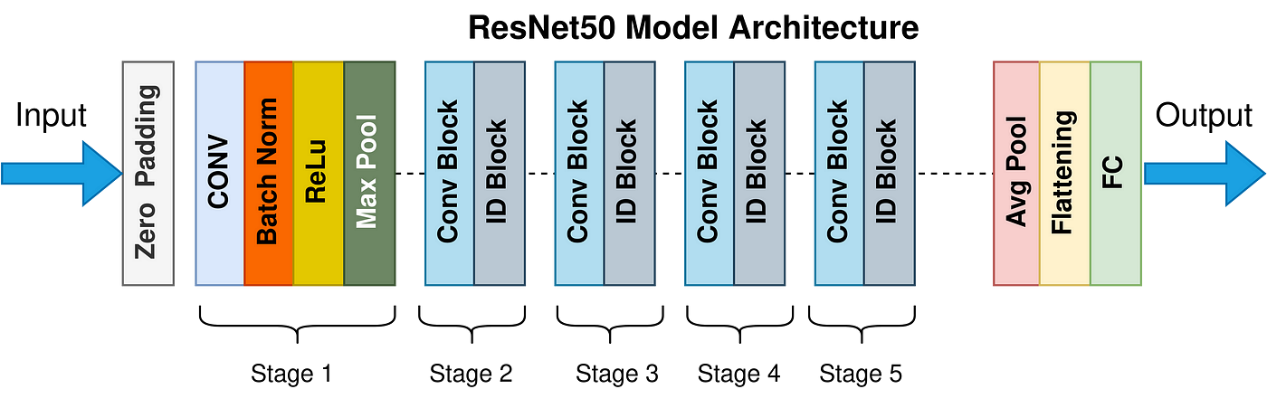


Figure 4: Architecture Diagram of ResNet50

### ResNet50:

ResNet50 complements the approach of learning intricate patterns with its deep architecture, mitigating the vanishing gradient problem and facilitating effective feature extraction.

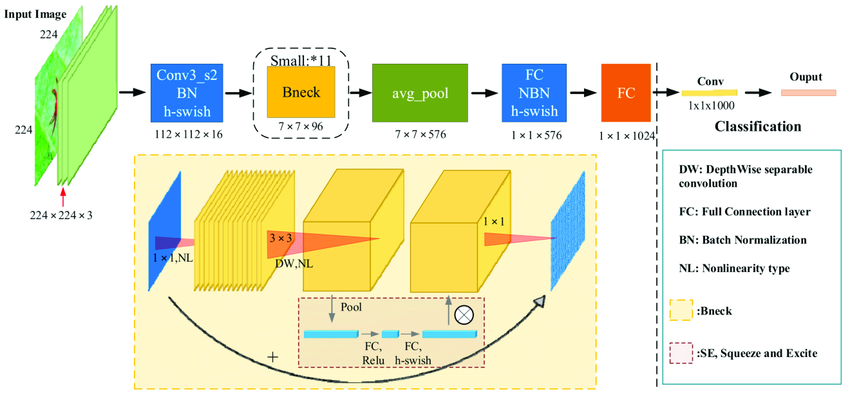


Figure 5: Architecture Diagram of MobileNetV3

### MobileNet:

MobileNetV3 is a computer vision model open-sourced by Google and designed for training classifiers. It uses depth wise convolutions to significantly reduce the number of parameters compared to other networks, resulting in a lightweight deep neural network.

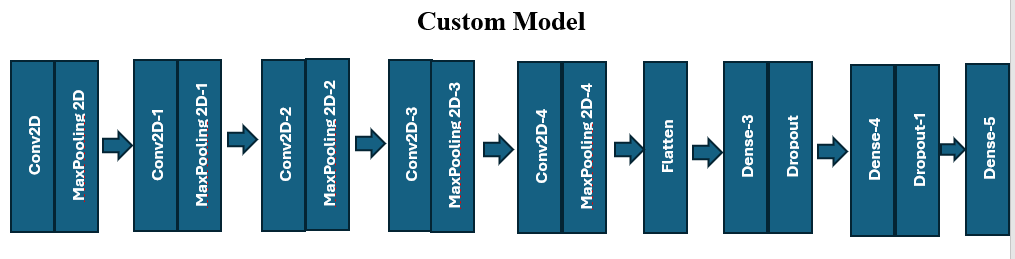


Figure 6: Architecture Diagram of Custom Model

### Custom Model:

In our Custom Model we added four blocks of Convolution and Max-Pooling layers. Then a Flatten layer was added. To extract complex features two blocks of Dense Leyer along with dropout were added and in last a Dense layer again. The architecture of our model, which had an accuracy of 96%, is drawn above.

## System Constraints:

### Software Constraints:

* The system must be interoperable with other medical devices.
* Accurate classification of images into Pneumonia detected, COVID-19, or none.
* User-friendly interface with clear instructions and visual aids.

### Hardware Constraints:

* Accessibility from any device with a web browser and internet connection.

### Cultural Constraints:

* Simple and intuitive web interface catering to users of diverse cultural backgrounds.
* Documentation provided in English for global accessibility.

## Flow Diagram:

The system architecture diagram of "X-ray Image classification for Pneumonia & COVID-19" provides a visual representation of the hierarchical structure guiding its operations. Upon user upload of an X-ray image, the system seamlessly classifies it into one of three categories: Pneumonia, COVID-19, or normal. Concurrently, users can access the generated results and, when logged in, review any existing records associated with their account. This holistic depiction encapsulates the efficiency and user-centric design of the system, ensuring a seamless and intuitive experience for all stakeholders involved.

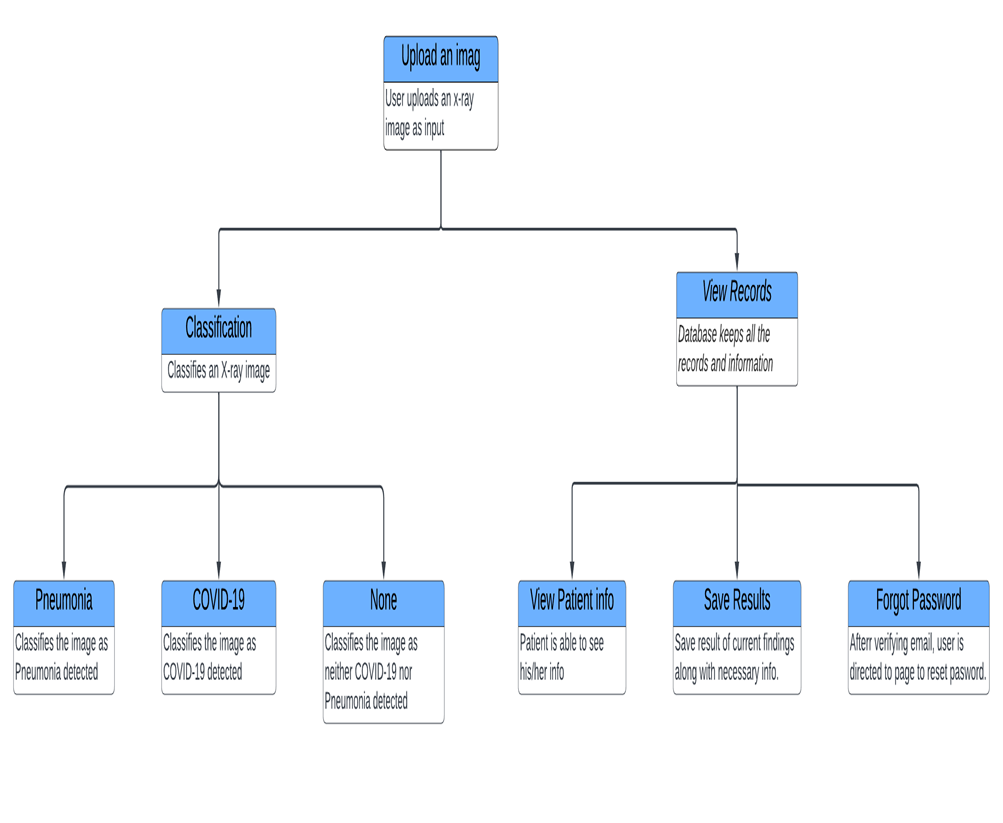


Figure 7: Flow Diagram

## Use Case Diagram:

In Figure, the Use Case Diagram elegantly illustrates the multifaceted interactions between users and the system. Users engage with the platform by uploading X-ray images, accessing results (via login authentication), logging in, signing up, or resetting forgotten passwords. The system seamlessly authenticates users, granting access to result viewing based on their login credentials. This comprehensive depiction encapsulates the fluidity and sophistication of user-system interactions, ensuring a seamless and secure experience for all users.

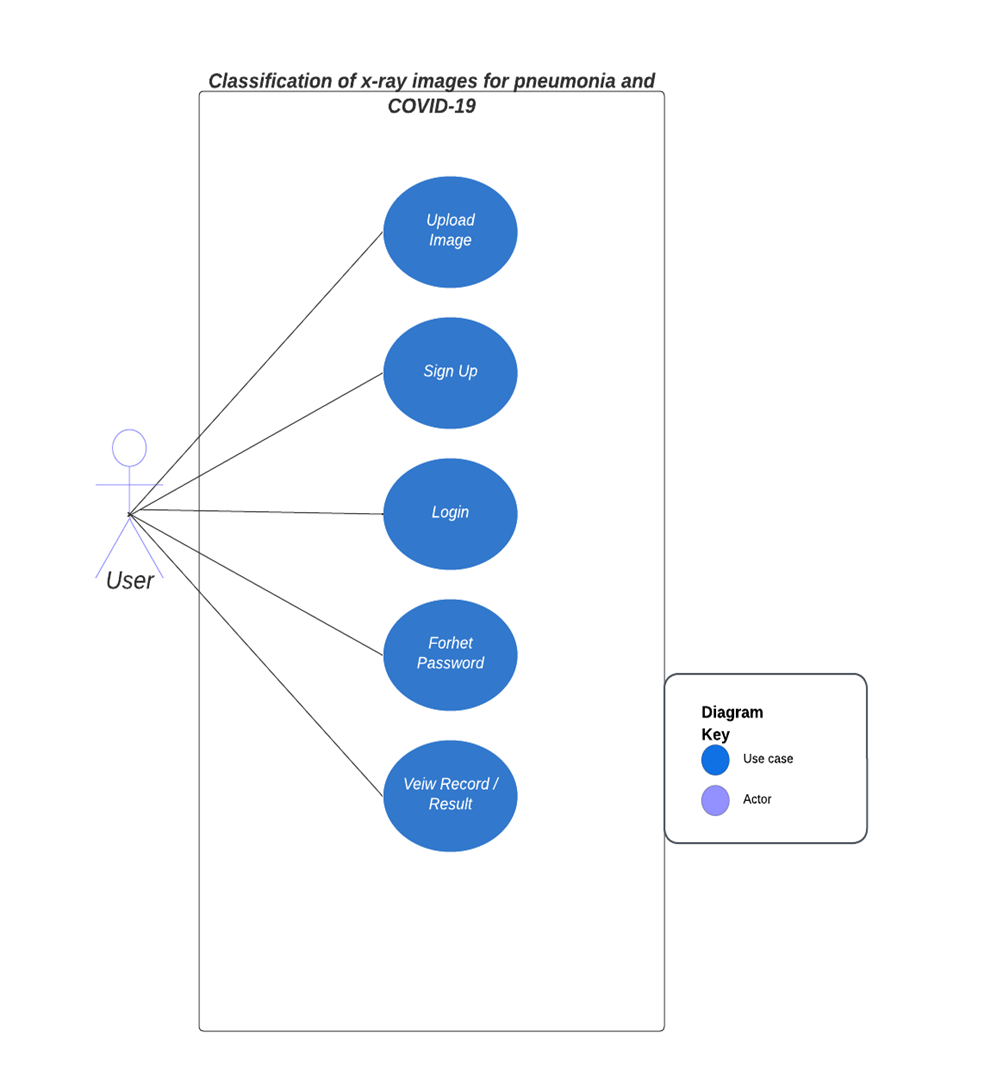
****

Figure 8: Use Case Diagram

## Sequence Diagram:

The Sequence Diagram offers a captivating glimpse into the seamless flow of interactions between users and the system. Users are empowered to upload images effortlessly, even without logging in initially. However, to access the insightful results, a sign-in is required. Upon image upload, users are seamlessly guided through the process of creating an account if they're not already registered. Subsequently, the system diligently verifies user authentication status. If not logged in, users are prompted to either sign in or create an account, fostering a secure and personalized experience. Upon successful sign-in, the system retrieves image results from the database, presenting them to users with effortless fluidity. This meticulous orchestration ensures the safeguarding of data integrity while facilitating a seamless and gratifying user experience.

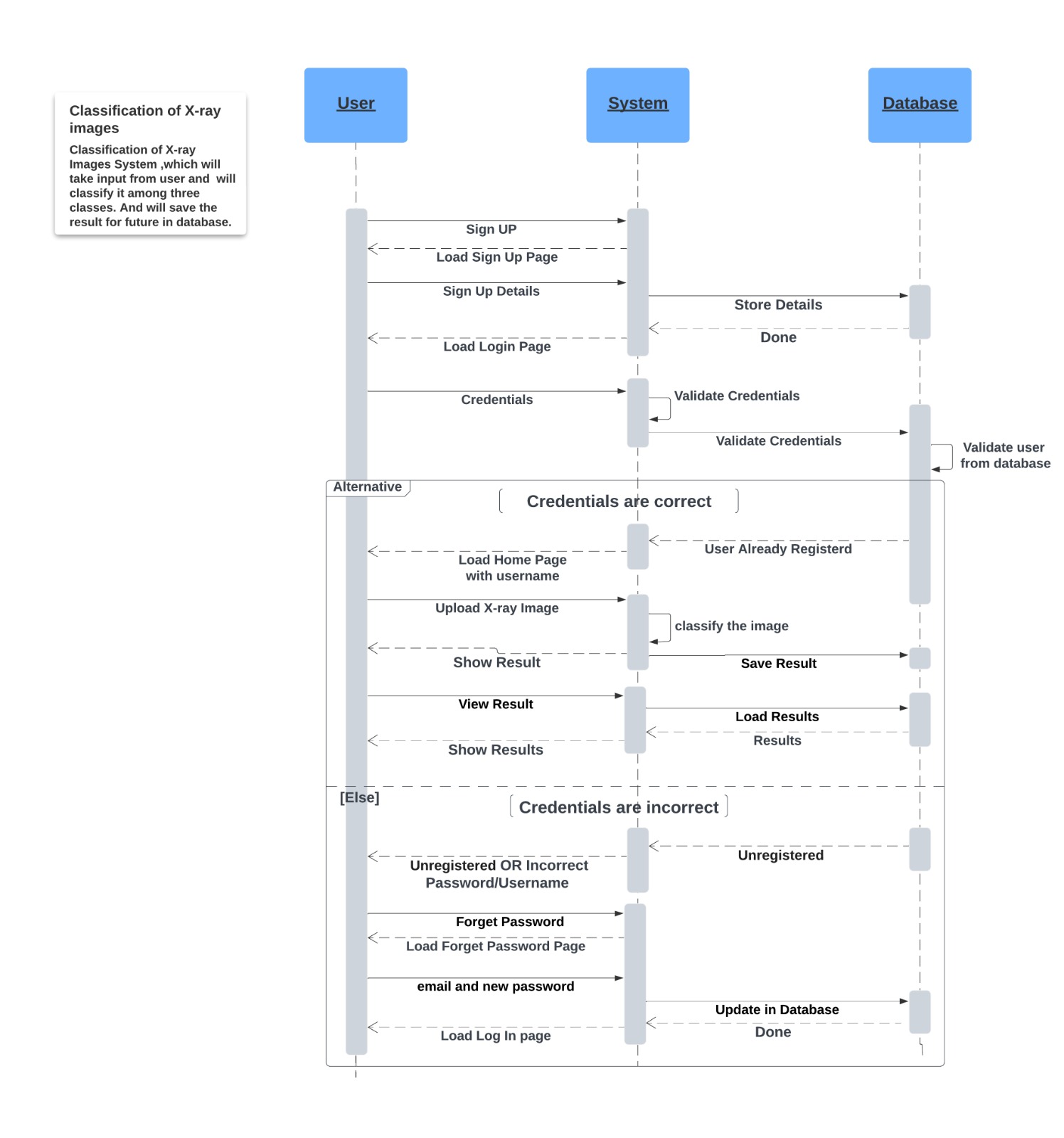


Figure 9: Sequence Diagram

## Entity Relationship Diagram:

The figure showcases the Entity Relationship diagram of our proposed system. It illustrates how user data is organized, with details stored in the User table, while corresponding user records are meticulously archived within the Records table. This diagram offers a clear visual representation of the system's data structure, ensuring efficient management and seamless retrieval of user information.

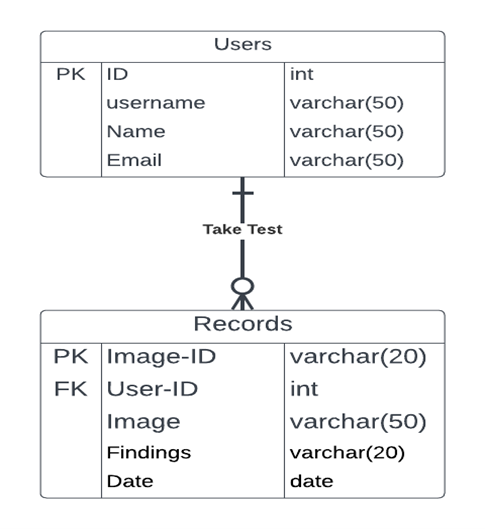
****

Figure 10: Entity Relationship Diagram

## Interface Design:

The Interface design unveils the welcoming Home page of our Web application, dedicated to "X-ray Image Classification for Pneumonia & COVID-19". Here, a user-friendly navbar guides exploration to key destinations: the About section, detailing the system's creators, and Records, offering access to personal records. The Home page itself presents two clear buttons: Upload image for submitting chest X-ray images and Show Results for viewing diagnoses. This straightforward layout ensures effortless navigation, embodying simplicity, and functionality for users of all backgrounds.

**A person holding an x-ray

Description automatically generated**

Figure 11: Home Page

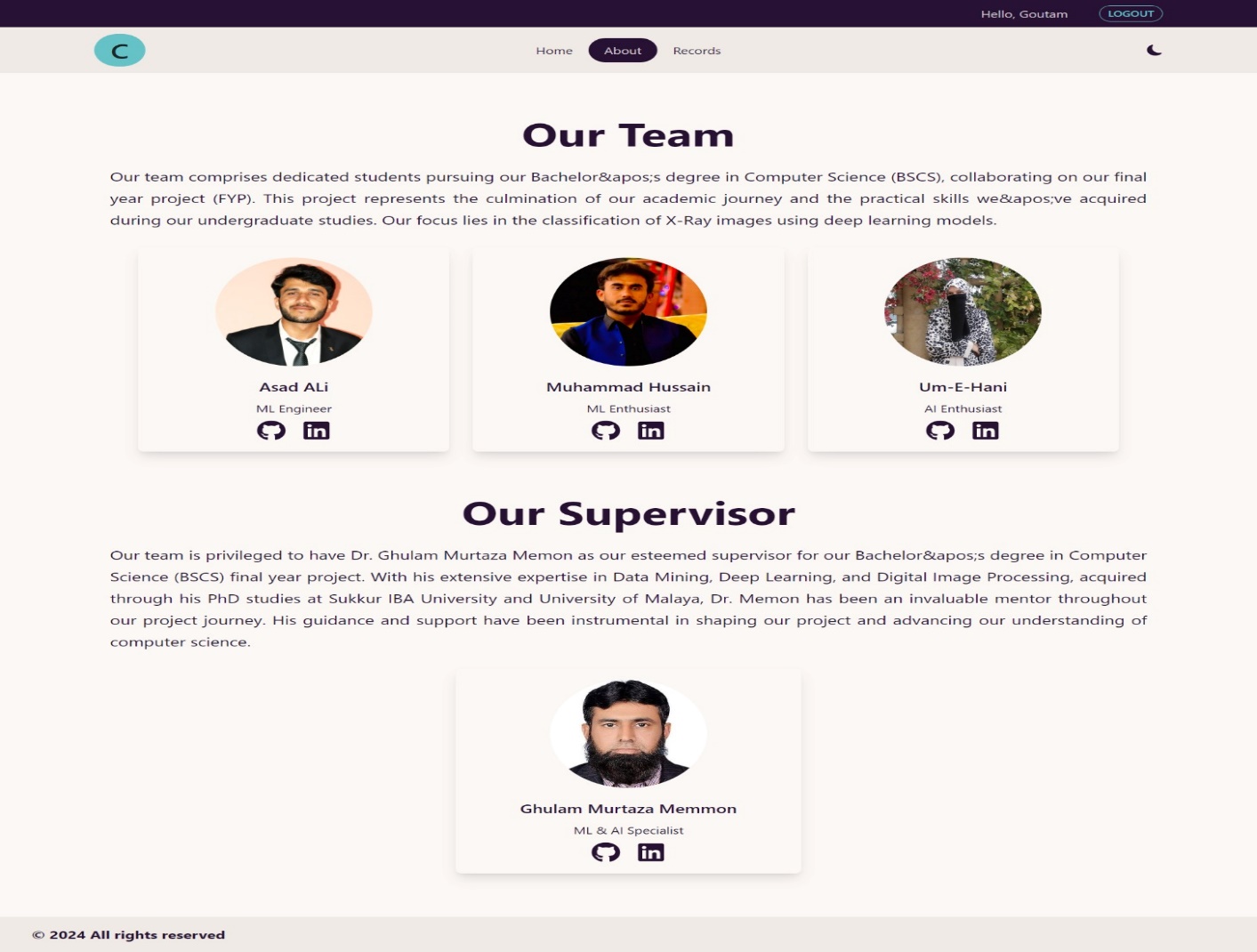
****

Figure 12: About Us Page

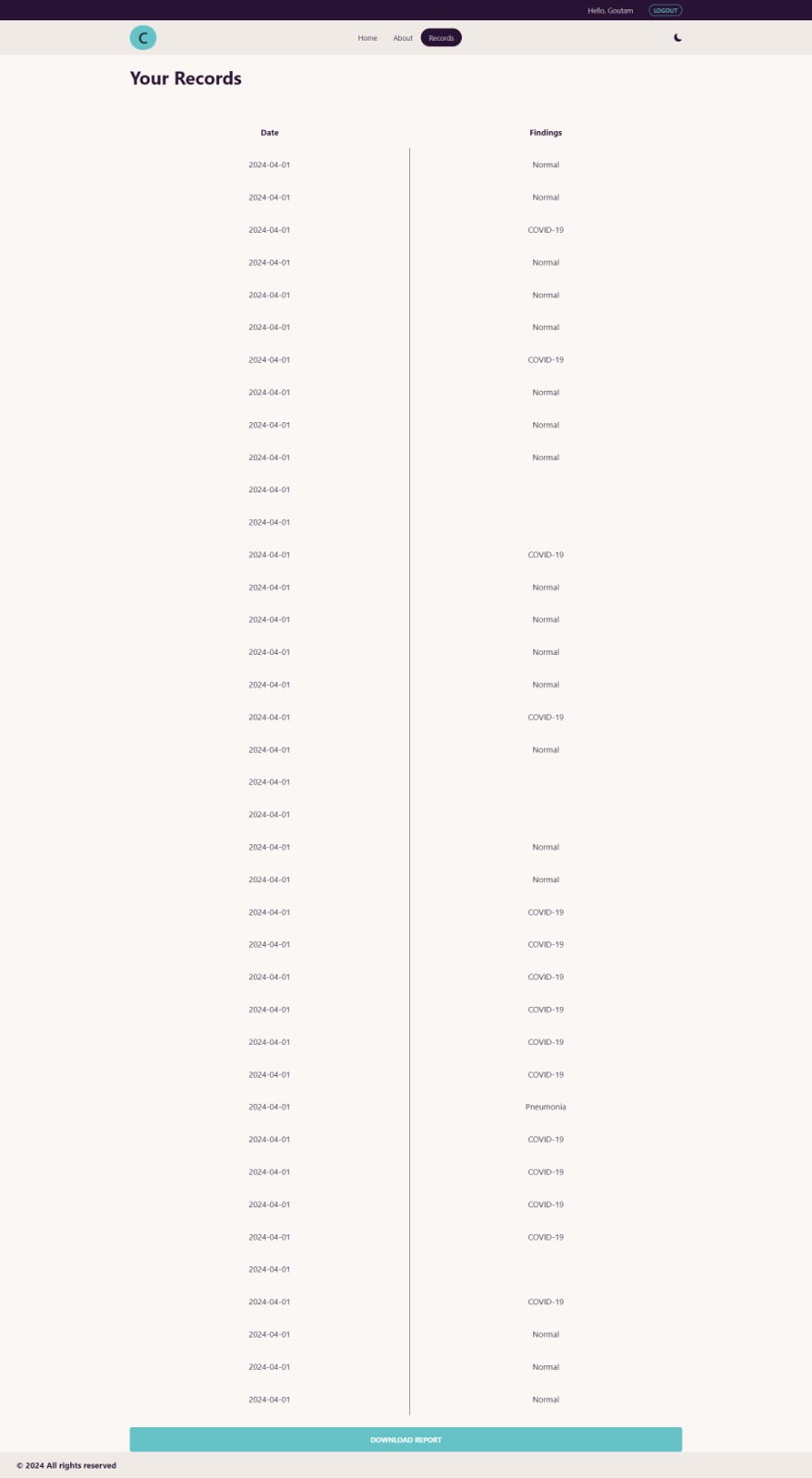
****

Figure 13: Records Page

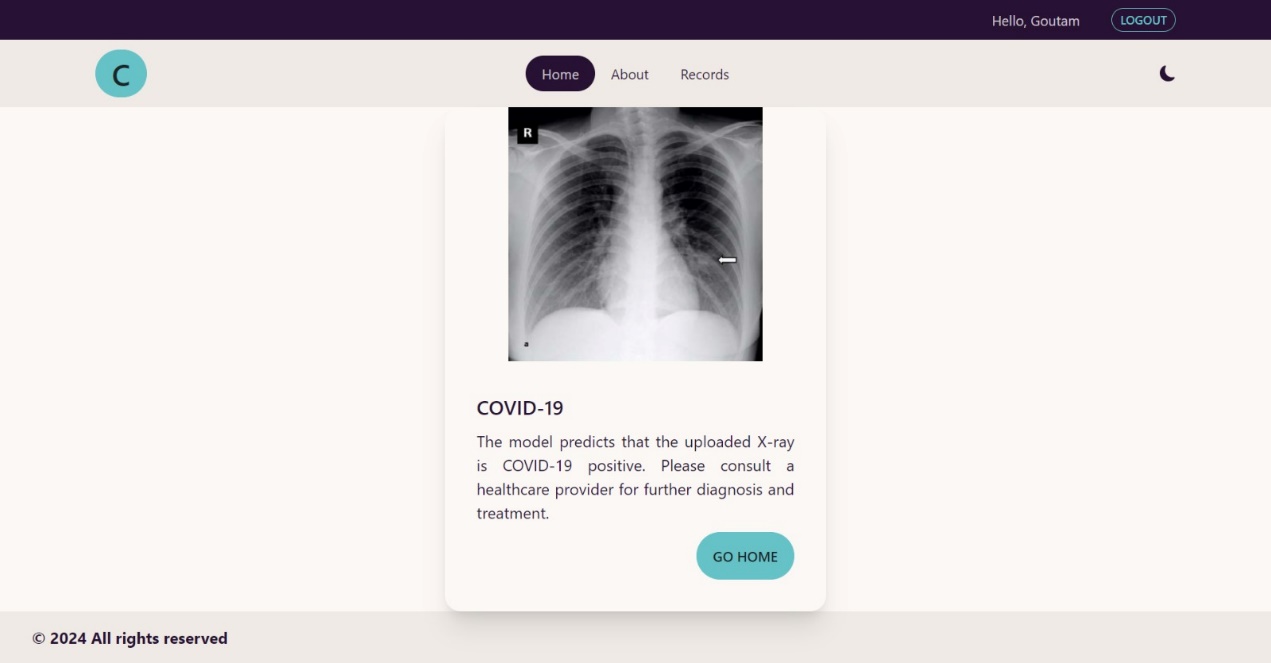
****

Figure 14: Covid Positive Case**A screenshot of a computer

Description automatically generated**Figure 15: Normal Case**A screenshot of a computer

Description automatically generated**

Figure 16: Pneumonia Positive

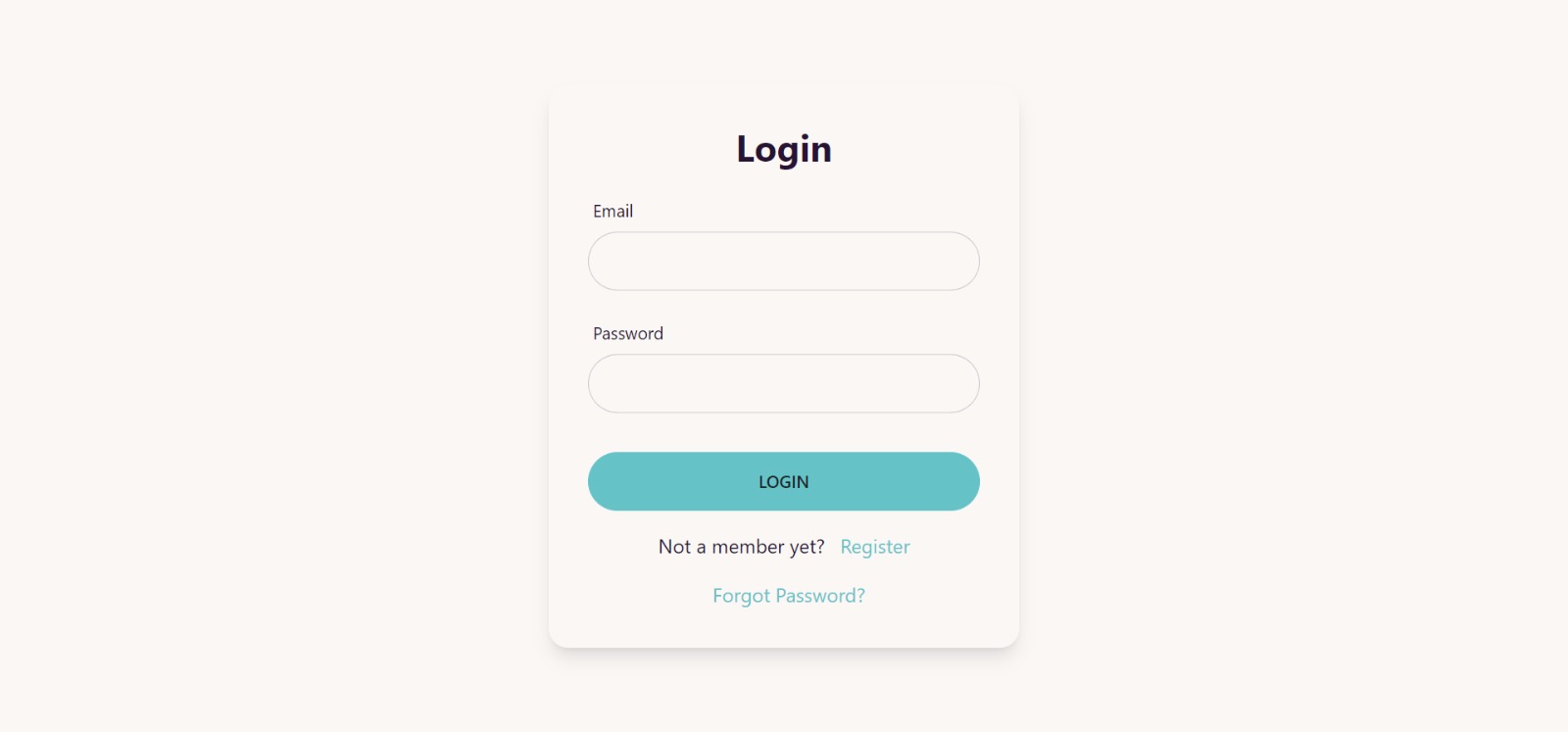
****

Figure 17: Login Page

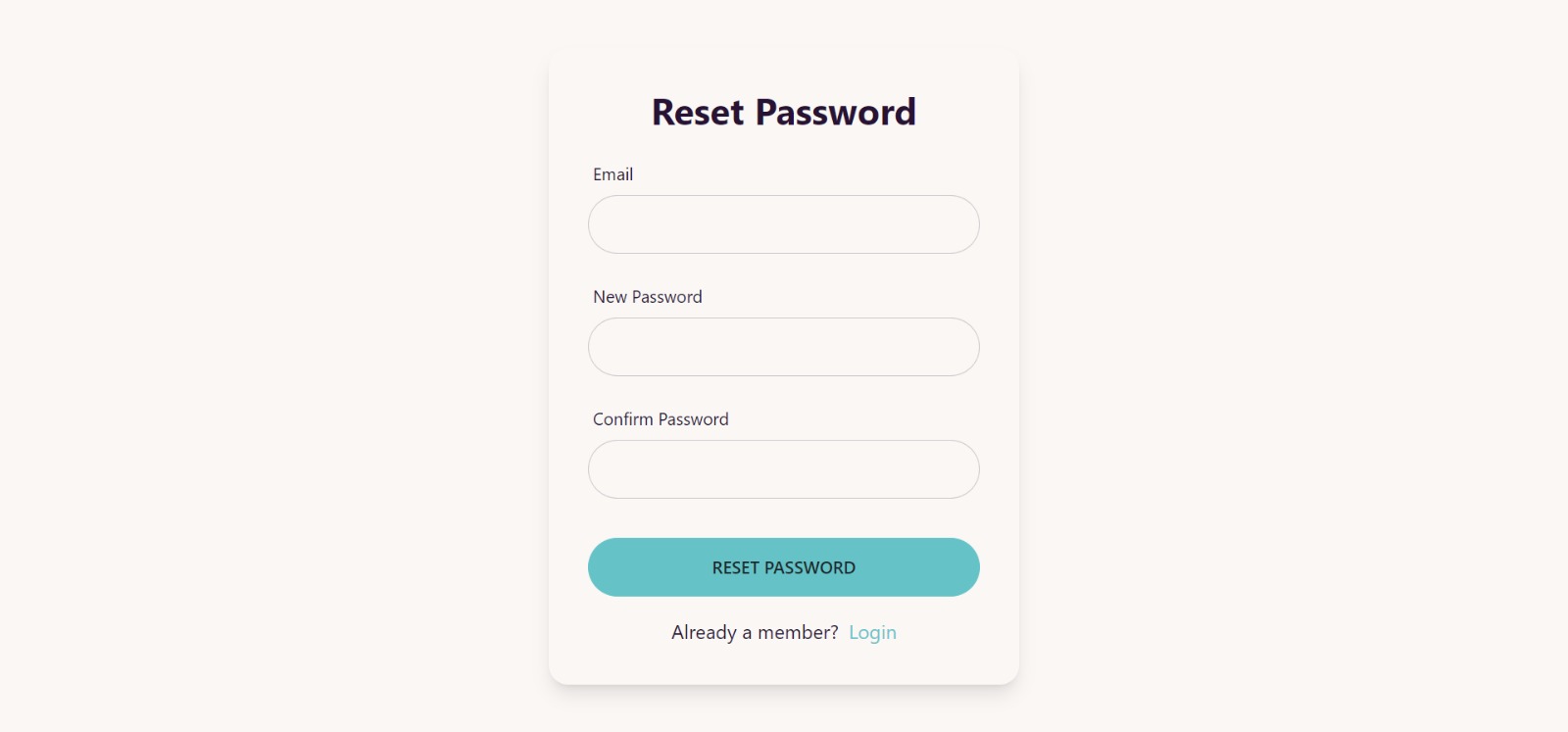
****

Figure 18: Reset Password Page

**A screen shot of a login form

Description automatically generated**

Figure 19: Register Page

# Implementation and Testing

The dataset comprises of total 5228 images, 3 different classes each with different number of images as following,

COVID: 1626 images

NORMAL: 1802 images

PNEUMONIA: 1800 images

## Version 0:

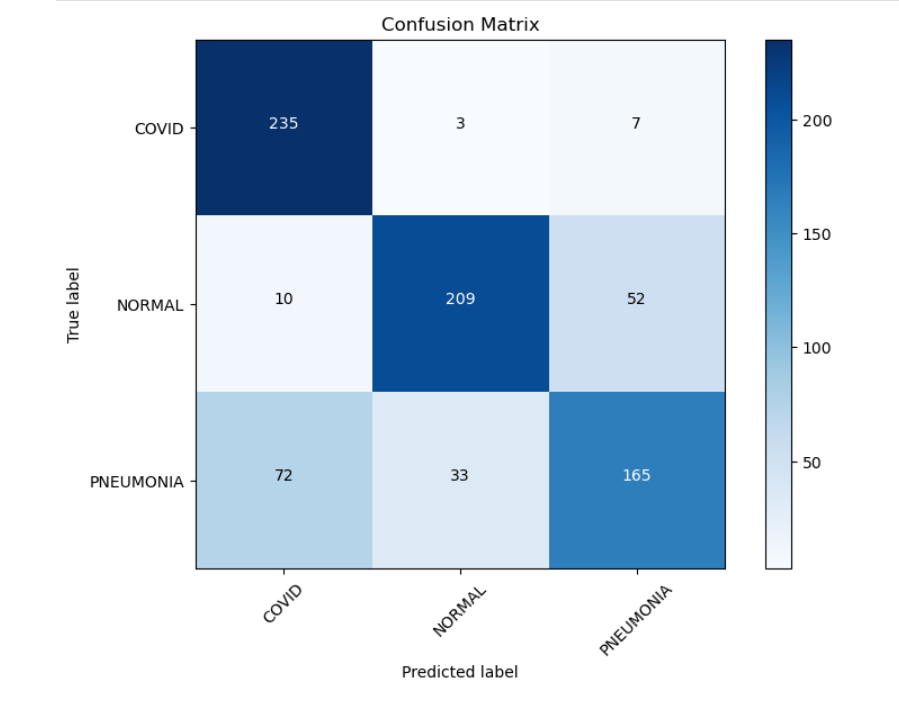


Figure 20: Confusion Matrix Version 0

The confusion matrix highlights discrepancies in predicting labels for Normal and Pneumonia instances, indicating potential misclassifications. Specifically, there are notable inaccuracies in distinguishing between Normal and Pneumonia cases.

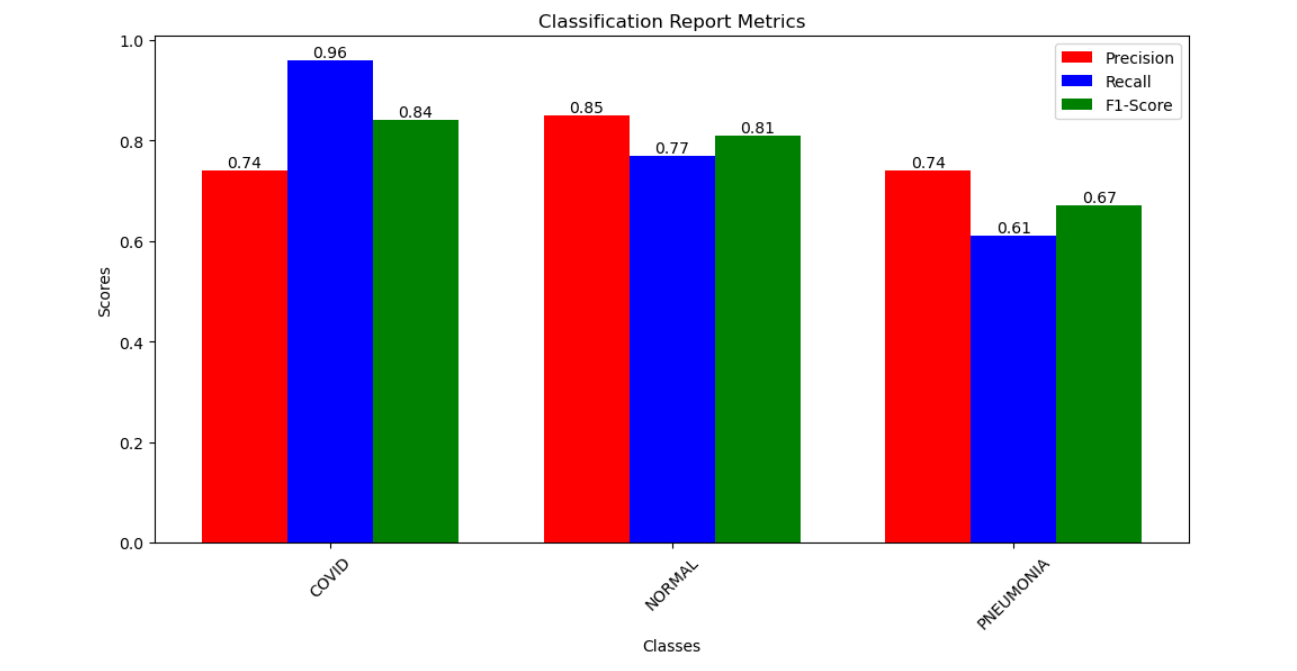


Figure 21: Classification Report Version 0

The classification report in figure, is providing insights into the precision, recall, and F1-score of each class in the prediction. For the classes "Covid," "Normal," and "Pneumonia," the report reveals the precision, recall, and F1-score for each. It assesses the accuracy of the model's predictions, highlighting any disparities in performance across different classes.

For COVID instances the precision is 0.74, Recall = 0.96 and F1 score is 0.84.

For Normal instances the precision is 0.85, Recall = 0.77 and F1 score is 0.81.

For Pneumonia instances the precision is 0.74, Recall = 0.61 and F1 score is 0.67.

## Version 1:

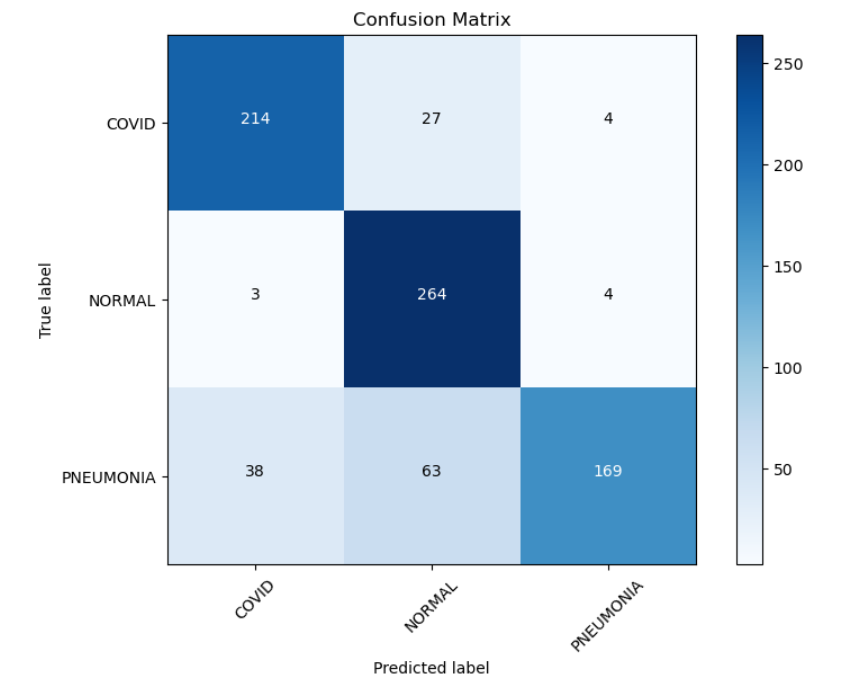
****

Figure 22: Confusion Matrix Version 1

The testing results showed improvement compared to previous models, suggesting progress in the model's performance. However, there were still instances where pneumonia cases were incorrectly classified as either COVID or normal, indicating ongoing challenges in accurately identifying pneumonia. Additionally, new issues emerged, particularly the misclassification of COVID cases as normal, highlighting the need for further refinement in the model's predictive capabilities.

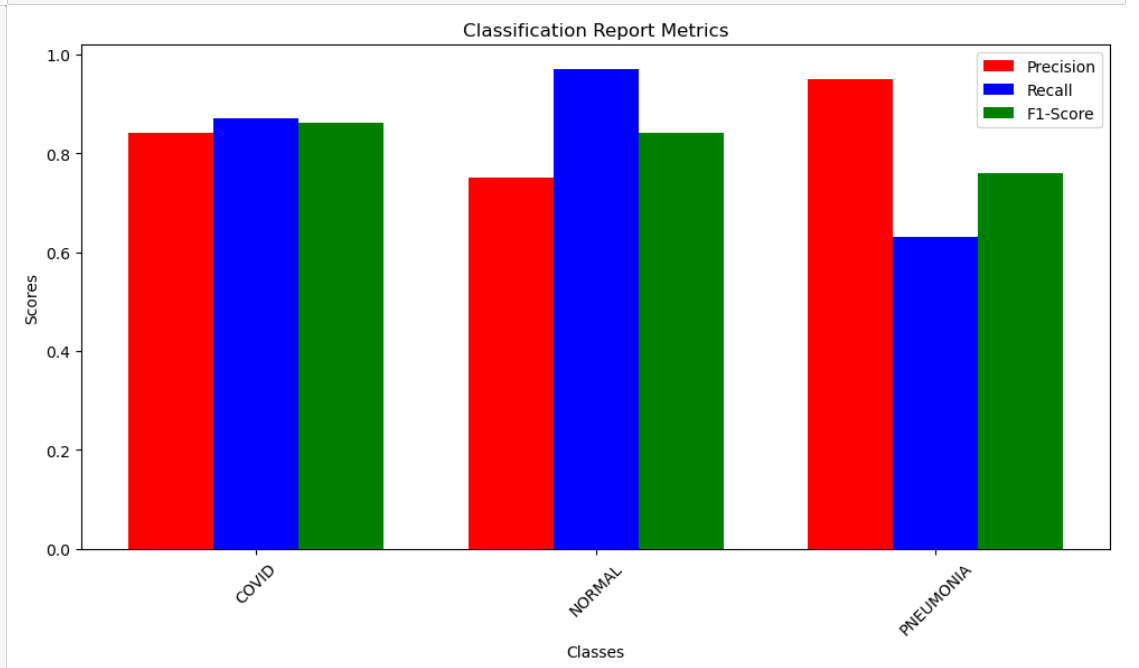


Figure 23: Classification Report Version 1

The classification report in figure, is providing insights into the precision, recall, and F1-score of each class in the prediction. For the classes "Covid," "Normal," and "Pneumonia," the report reveals the precision, recall, and F1-score for each. It assesses the accuracy of the model's predictions, highlighting any disparities in performance across different classes.

## Version 2:

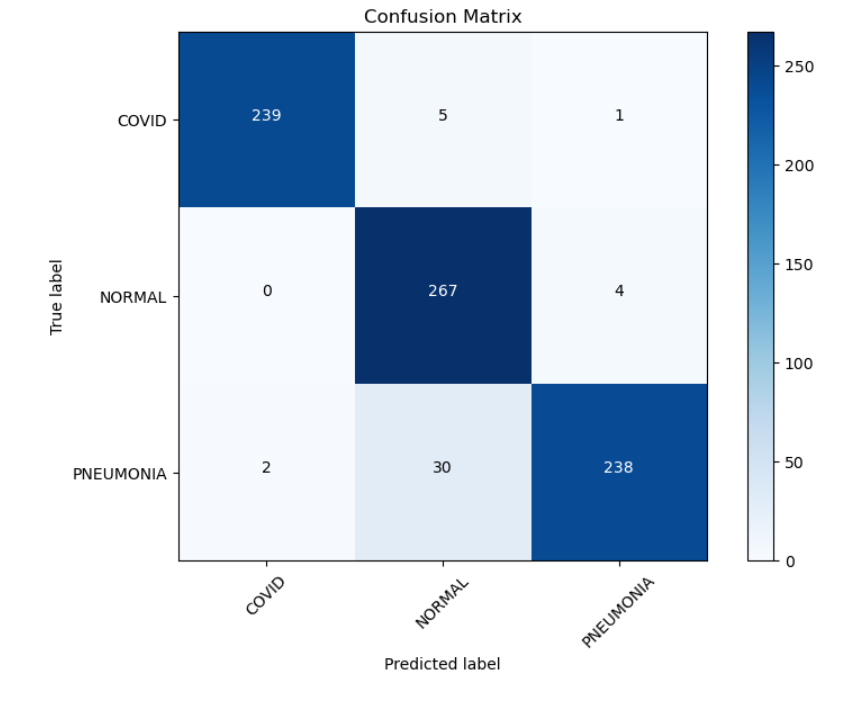
****

Figure 24: Confusion Matrix Version 2

The enhancements implemented in the model led to a notable increase in accuracy, reaching an impressive 95%, demonstrating significant progress compared to earlier versions. Despite this improvement, a minor issue persisted where some normal instances were still not predicted correctly, indicating the need for additional investigation, and fine-tuning to address this specific misclassification.

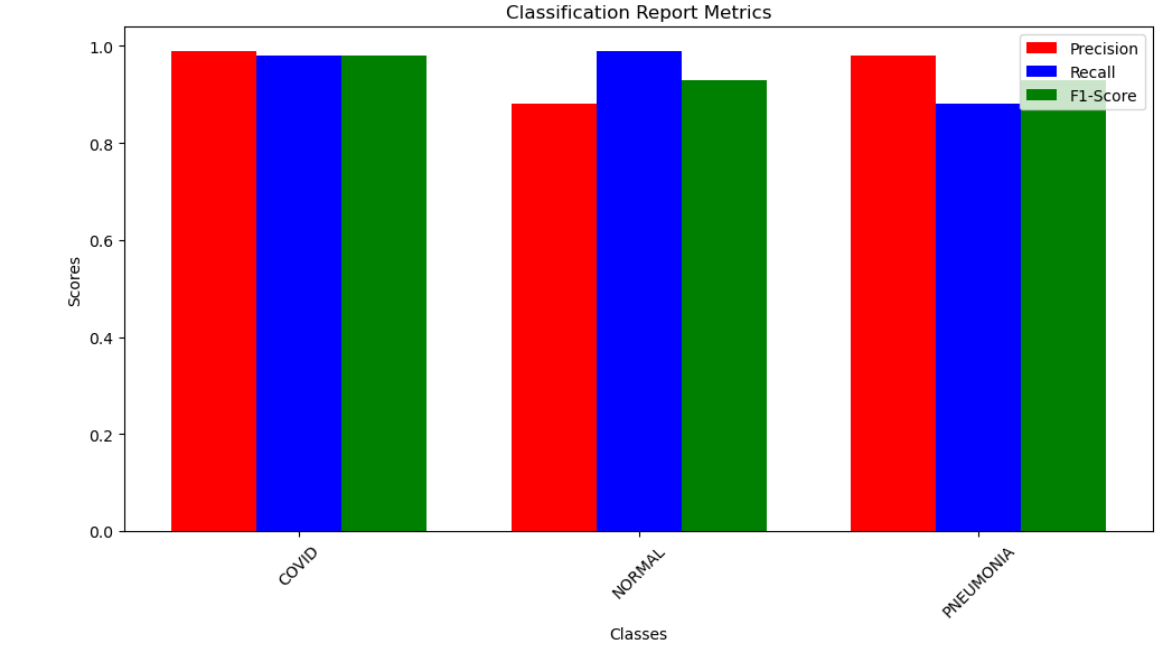


Figure 25: Classification Report Version 2

The classification report in figure, is providing insights into the precision, recall, and F1-score of each class in the prediction. For the classes "Covid," "Normal," and "Pneumonia," the report reveals the precision, recall, and F1-score for each. It assesses the accuracy of the model's predictions, highlighting any disparities in performance across different classes.

## Version 3:

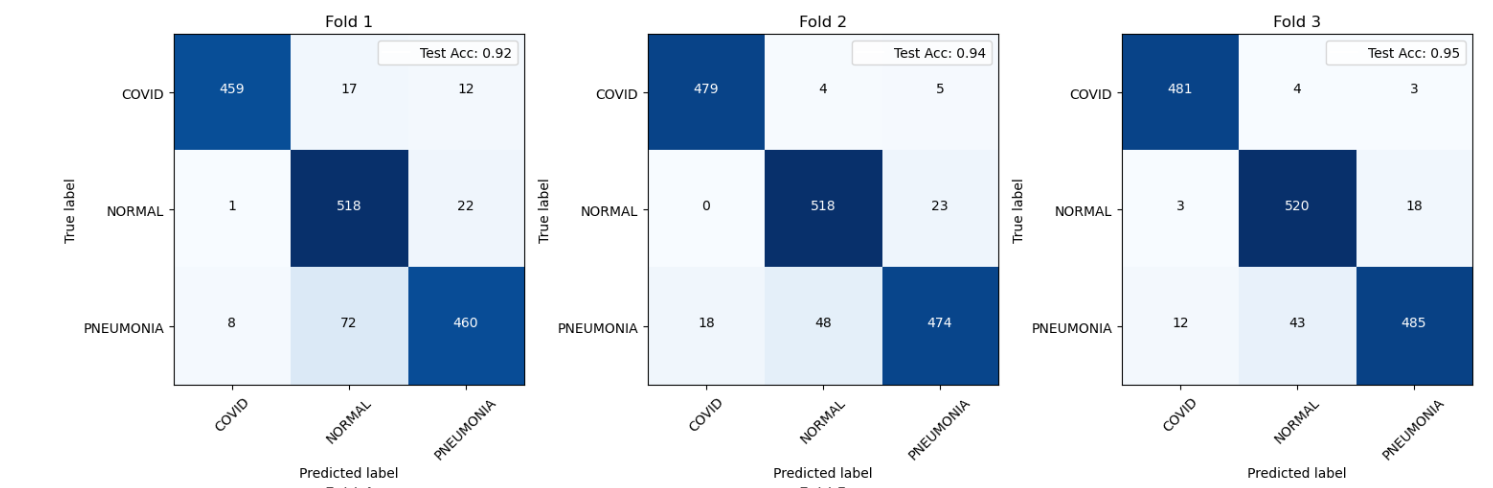
****

Figure 26: Confusion Matrices for Folds 1,2,3 of Version 3

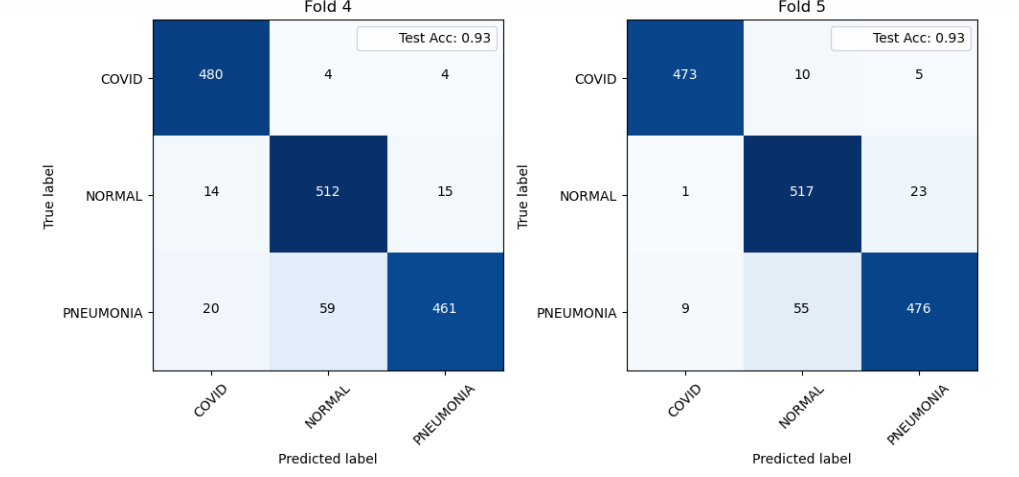
****

Figure 27: Confusion Matrices for Fold 4,5 of Version 3

The data was divided into 5 folds to enable cross-validation, ensuring robustness and preventing overfitting of the model. Among these folds, the third fold stood out by delivering the most impressive performance, boasting an accuracy of 95% as shown in Confusion Matrices. On average, across all folds, the model maintained a high accuracy level of 94%, underscoring its consistency and reliability across different subsets of the dataset.

## Version 4:

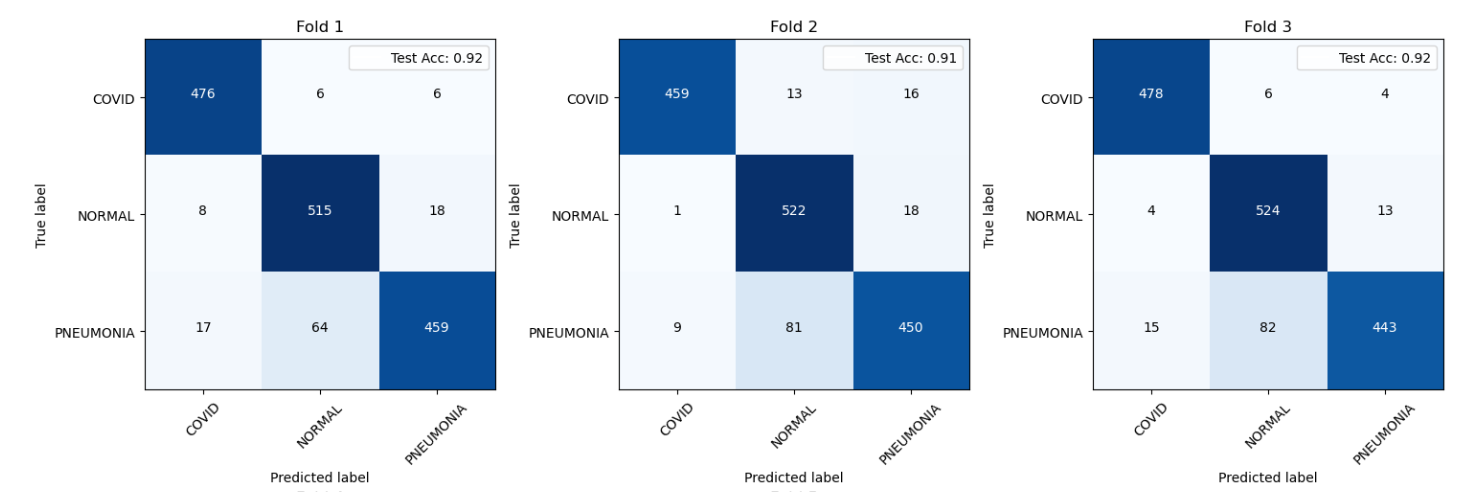
****

Figure 28: Confusion Matrix for Folds 1,2,3 for Version 4

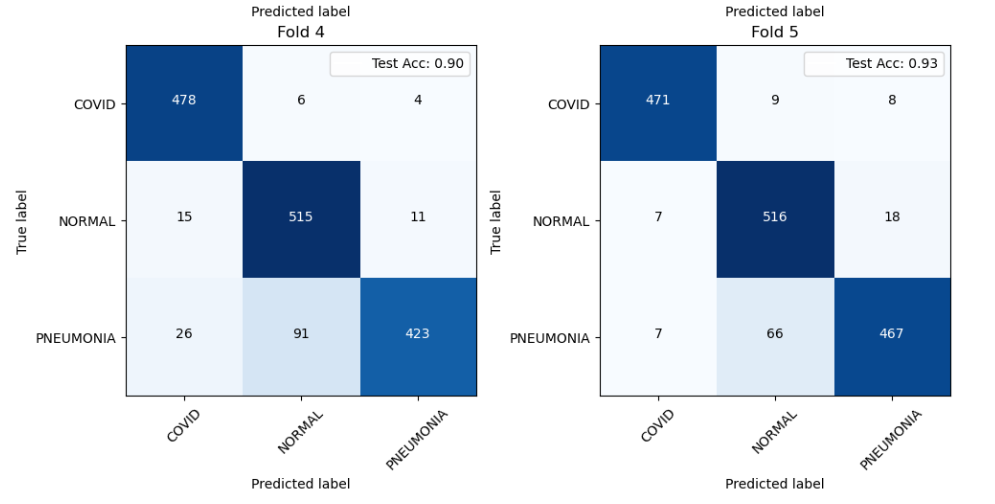


Figure 29: Confusion Matrices for Folds 4,5 for Version 4

Through the utilization of 5-fold cross-validation, our model achieved its peak accuracy of 93% on fold-5 as shown in Confusion Matrices.

## Version 5:

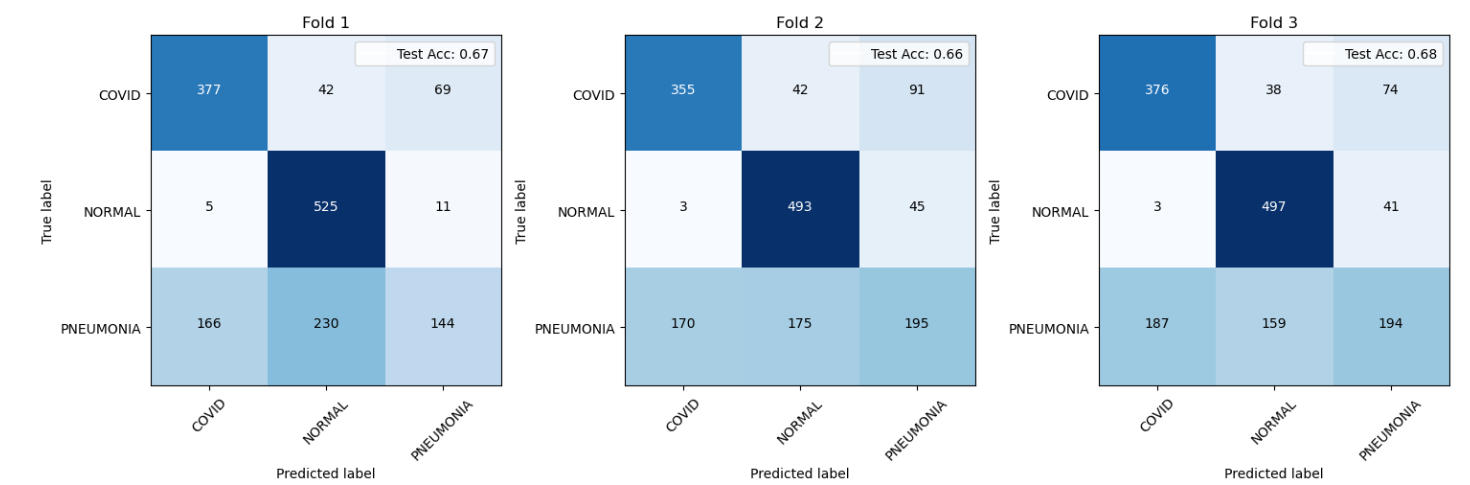
****

Figure 30: Confusion Matrices for Folds 1,2,3 for Version 5

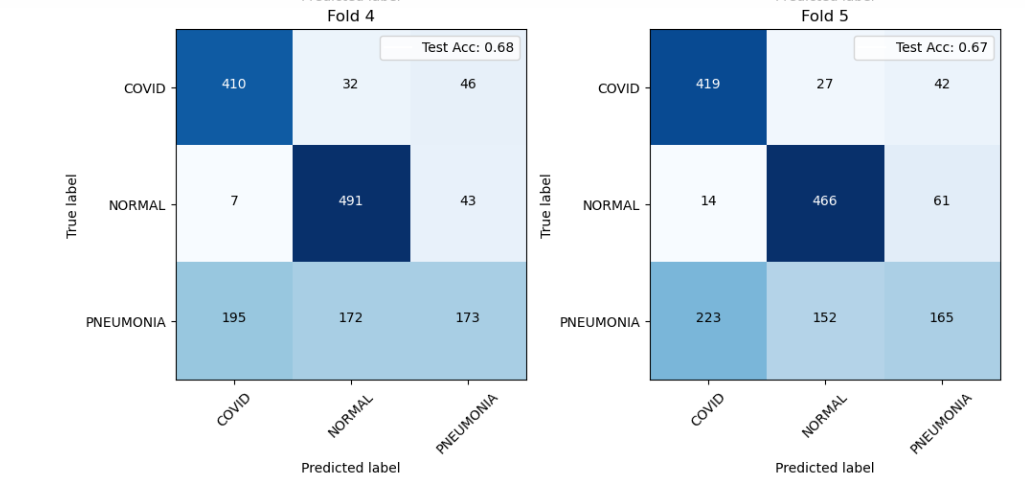


Figure 31: Confusion Matrices for Folds 4,5 for Version 5

In this scenario, employing ResNet50 within a 5-fold cross-validation framework yielded a maximum accuracy of 68% as can be seen in Confusion Matrices above. However, the significant misclassification of pneumonia instances suggests a limitation in the available dataset for effectively training a complex model like ResNet50. While keeping other parameters consistent, further investigation and potential adjustments are warranted to rectify these misclassifications and improve the overall performance of the model in accurately identifying pneumonia cases.

## Version 6:

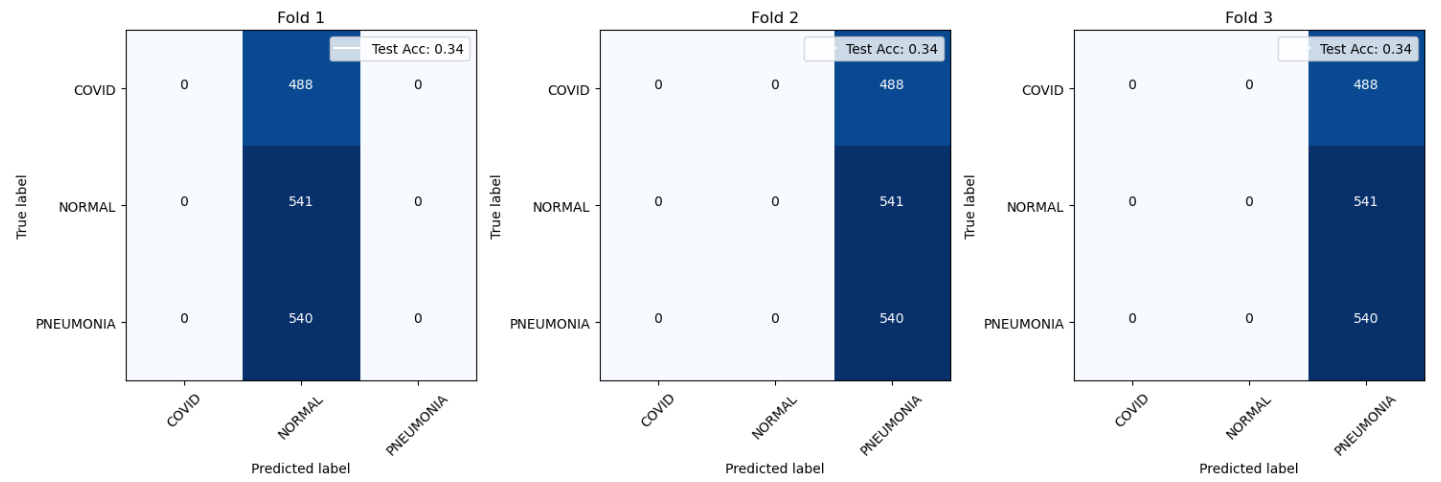
****

Figure 32:Confusion Matrices for Folds 1,2,3 for Version 6

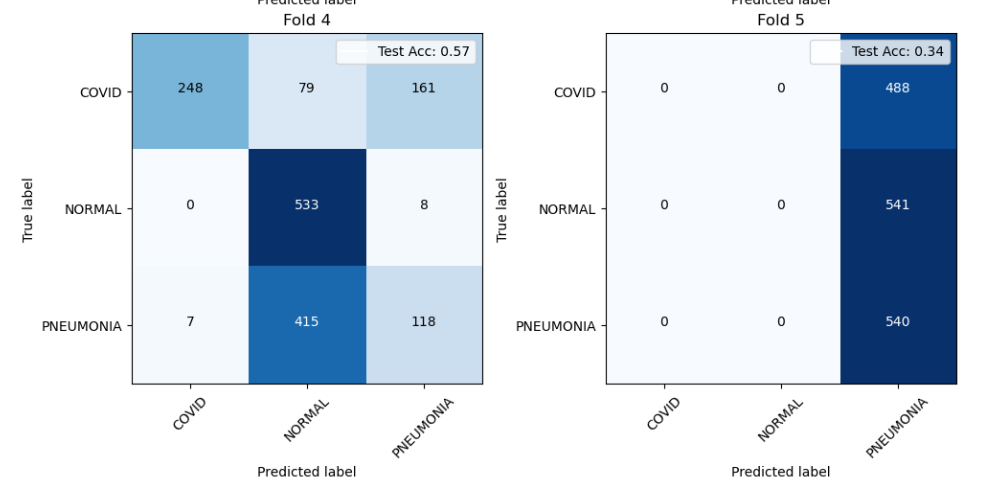


Figure 33: Confusion Matrices for Folds 4,5 for Version 6

During fold 4 of cross-validation, MobileNet demonstrated its efficacy by achieving a peak accuracy of 57%, underscoring its ability to effectively classify instances within the dataset.

## Version 7:

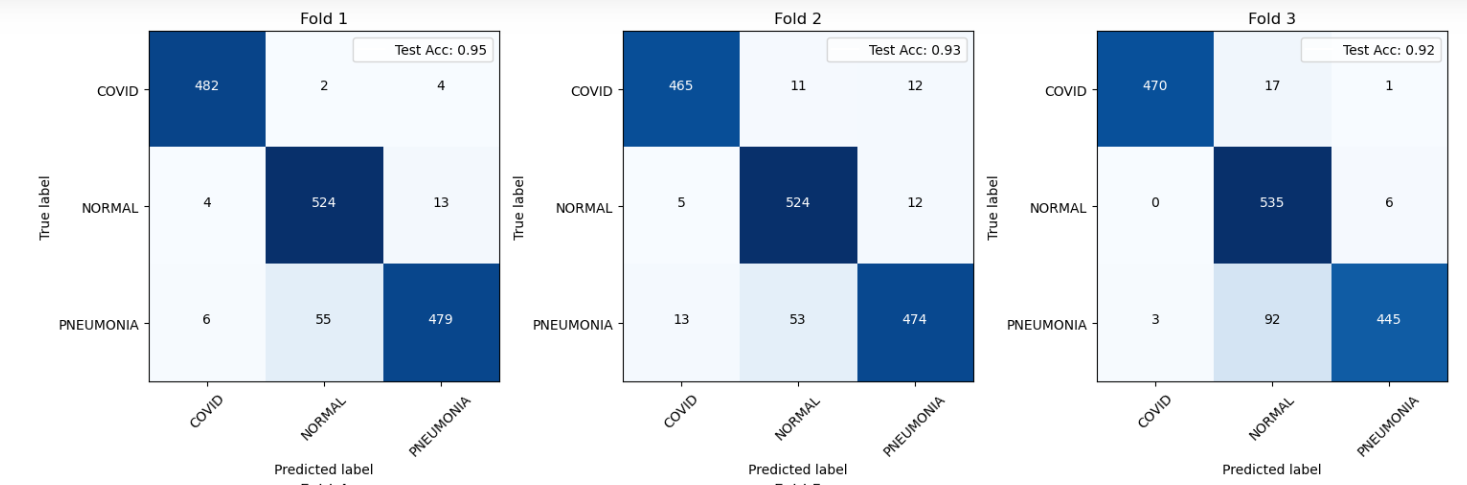
****

Figure 34: Confusion Matrices for Folds 1,2,3 for Version 7



Figure 35: Confusion Matrices for Folds 4,5 for Version 7

Our customized model achieved its peak accuracy of 96% during fold 4 of cross-validation, highlighting its robustness in accurately classifying instances within the dataset.

# Results and Discussion

## Findings and Problems Version 0:

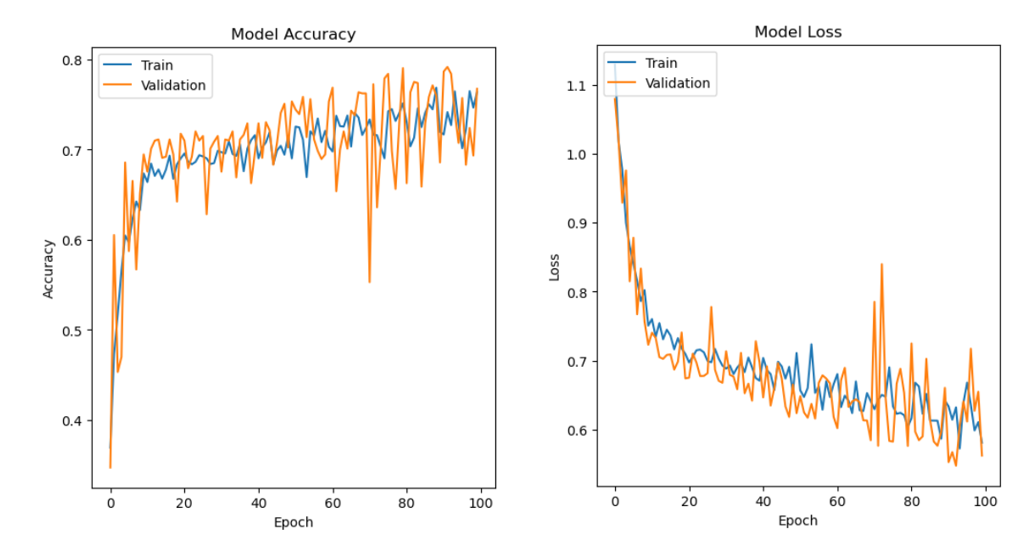


Figure 36: Accuracy & Loss Graphs Version 0

* The confusion matrix revealed inaccuracies in predicting labels for pneumonia and normal instances.
* The model's size posed challenges due to insufficient data for effective training.
* Accuracy fluctuated between 0.68 and 0.78 from the sixth epoch onwards.

## Findings and Problems Version 1:

****

Figure 37: Accuracy & Loss Graphs Version 1

* The model was modified(VGG-16 was used) in this version due to previous model’s high data requirements for training.
* Data augmentation techniques were implemented, and the number of epochs was reduced to 10.
* Testing results improved compared to previous models, indicating progress. However, pneumonia instances were still falsely predicted as compared to COVID or normal.
* New issues arose, notably the misclassification of COVID cases as normal.

## Findings & Problems Version 2:

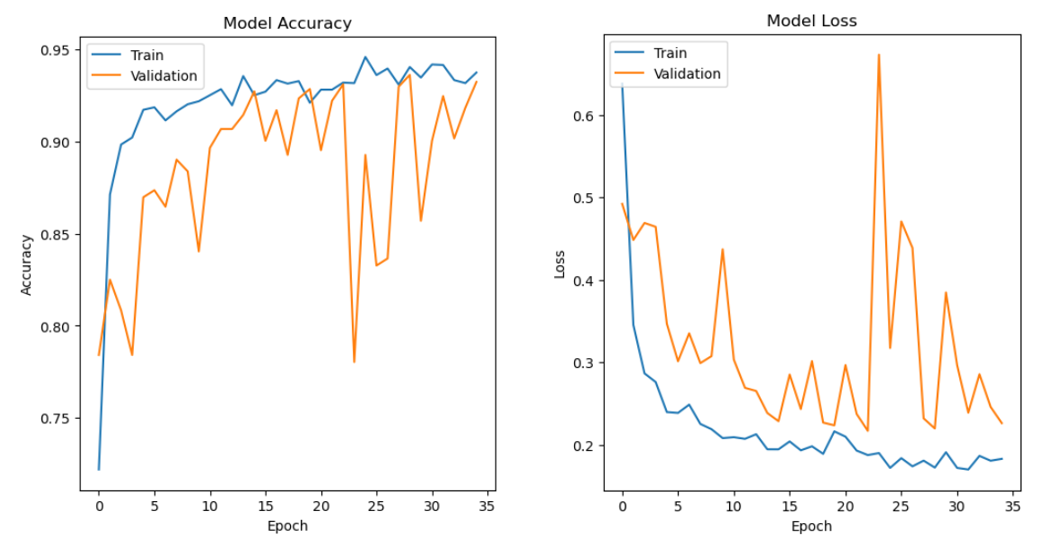
****

Figure 38: Accuracy & Loss Graphs Version 2

* Significant improvements were realized in version 2 of the model by integrating custom Dense layers into the VGG-16 architecture and employing the Adam optimizer.
* These enhancements propelled the accuracy to an impressive 95%, showcasing substantial progress over previous versions.
* However, a minor issue persisted as a few normal instances were not predicted correctly, warranting further investigation and refinement.

## Findings & Problems Version 3:

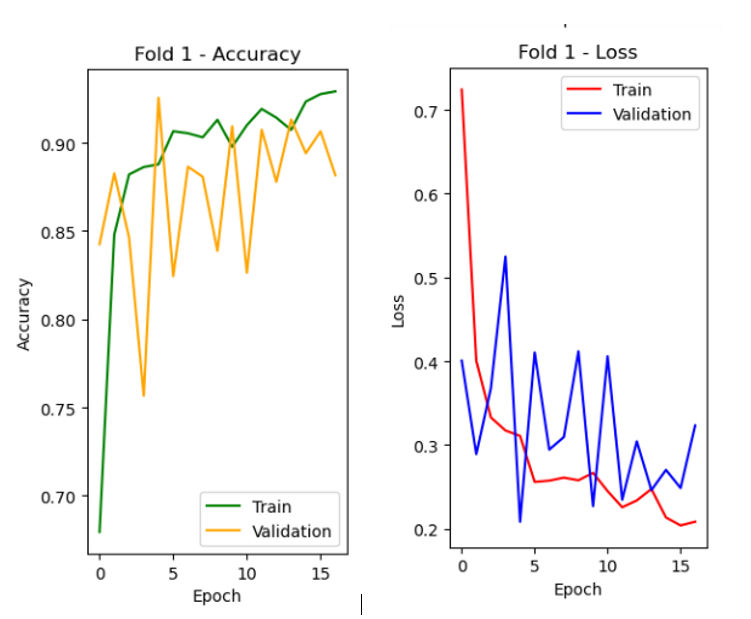
****

Figure 39: Accuracy & Loss fold 1 Version 3

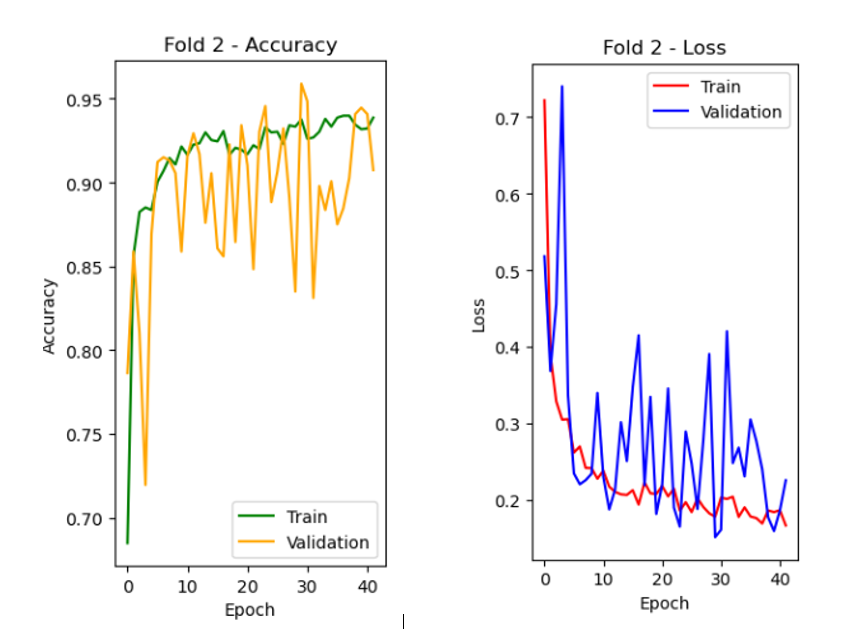
****

Figure 40: Accuracy & Loss fold 2 Version 3

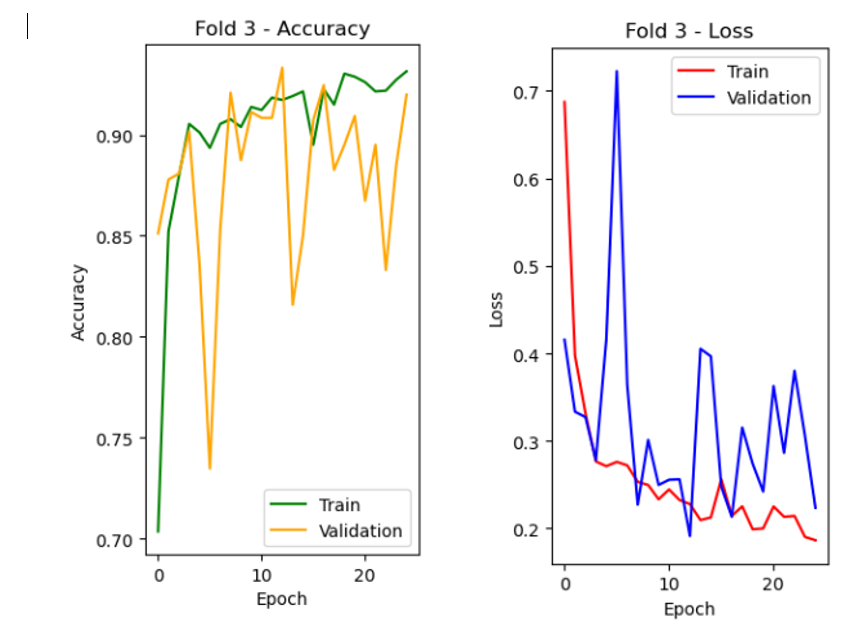
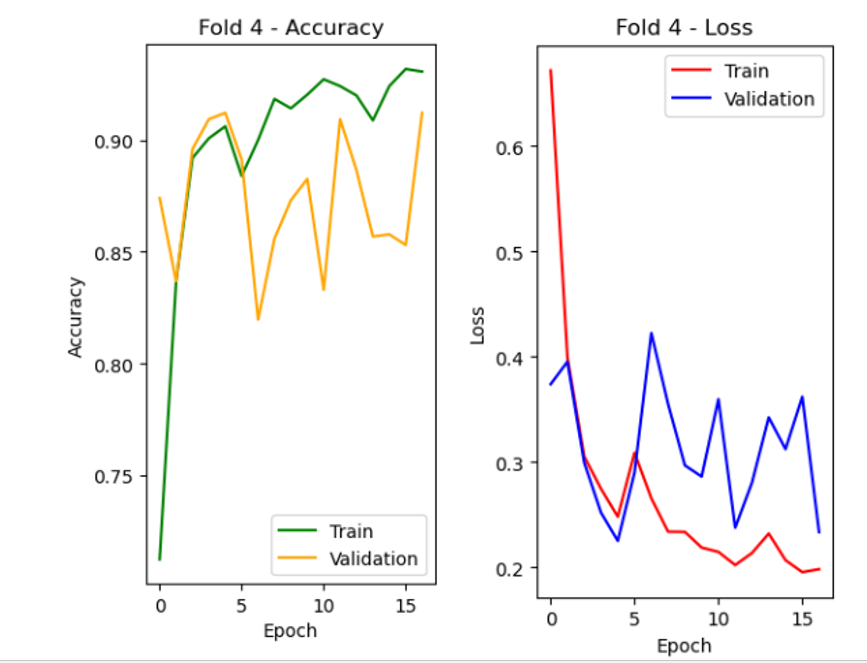
****Figure 42: Accuracy & Loss fold 3 Version 3

Figure 41: Accuracy & Loss fold 4 Version 3

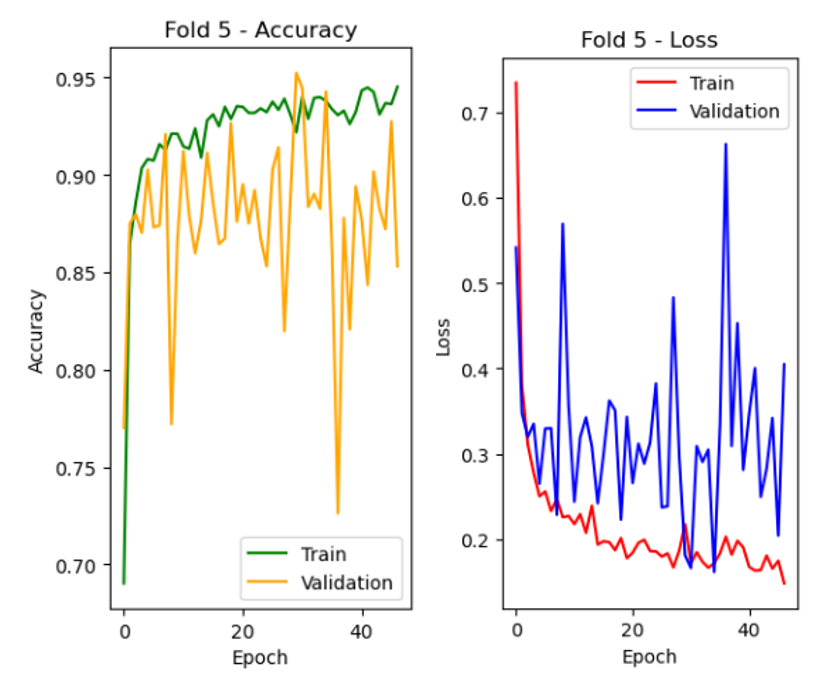
****

Figure 43: Accuracy & Loss fold 5 Version 3

* The data was partitioned into 5 folds to facilitate cross-validation and mitigate overfitting.
* Notably, the third fold yielded the best results, achieving a remarkable accuracy of 95%, with an average accuracy of 94% across all folds.
* The optimizer, learning rate, and additional Dense layers remained consistent throughout the validation process, ensuring uniformity in model evaluation.

## Findings & Problems Version 4:

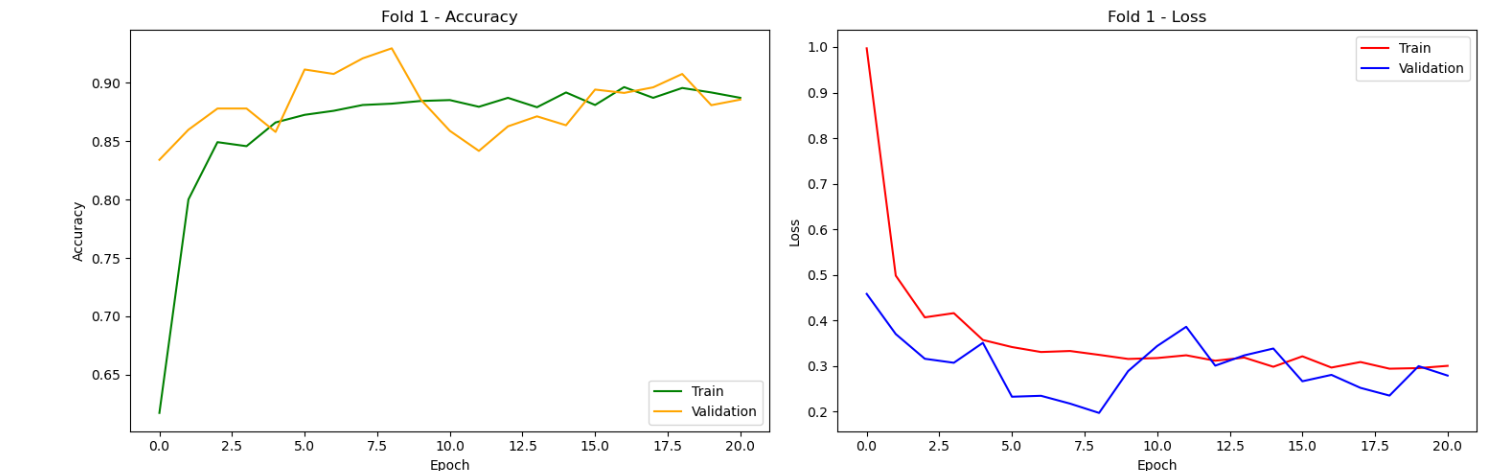
****

Figure 44: Accuracy & Loss fold 1 Version 4

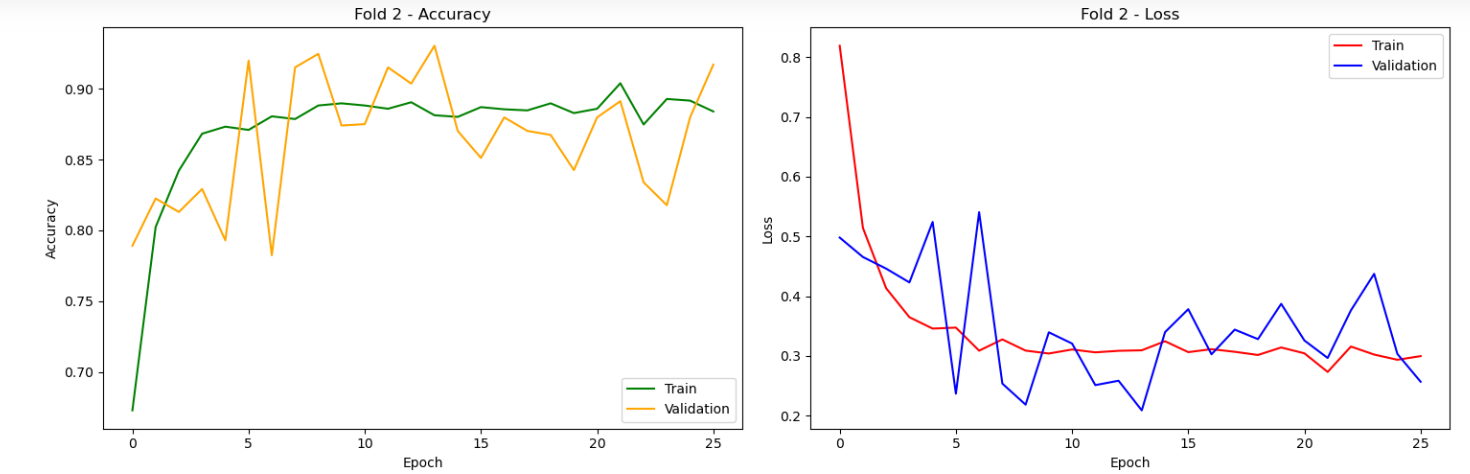


Figure 45: Accuracy & Loss fold 2 Version 4

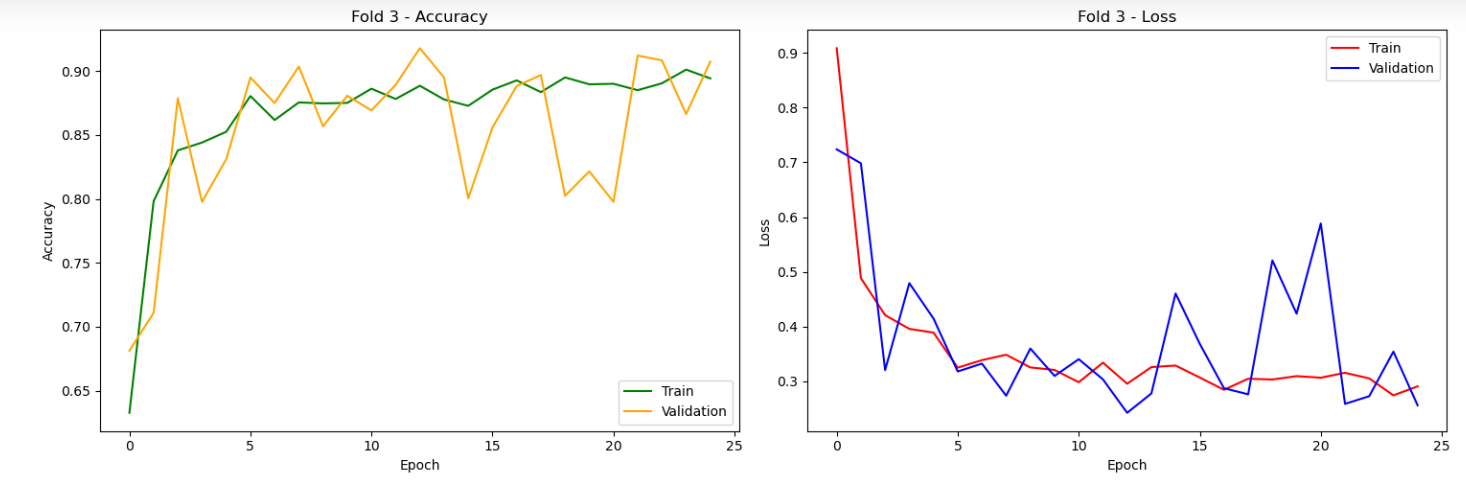


Figure 46: Accuracy & Loss fold 3 Version 4

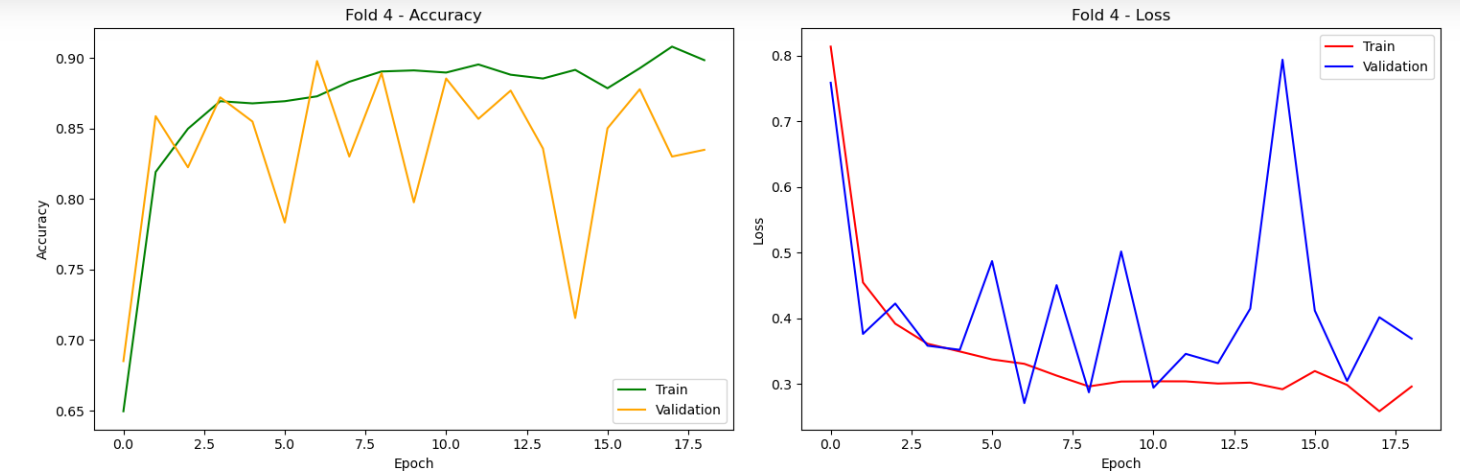


Figure 47: Accuracy & Loss fold 4 Version 4

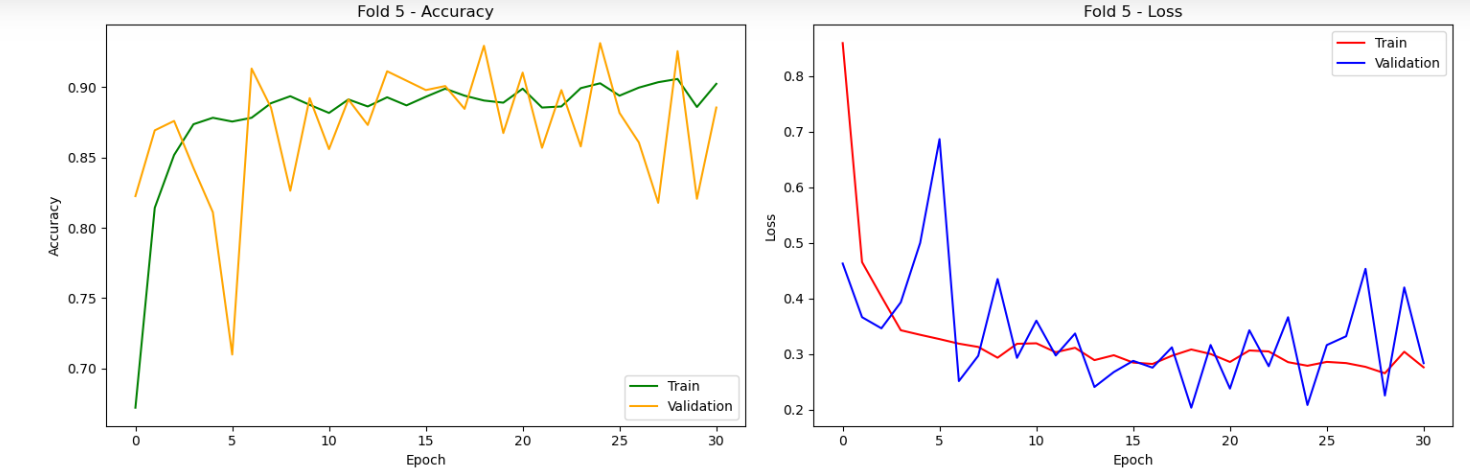


Figure 48: Accuracy & Loss fold 5 Version 4

* Employing 5-fold cross-validation, the highest accuracy of 93% was attained on fold-5, demonstrating robust model performance.
* Additionally, two additional dense layers, each comprising 4096 nodes, were incorporated, along with a dropout rate of 0.5 to mitigate overfitting.
* Despite being configured to train for 50 epochs, the implementation of the model\_early\_stopping parameter led to variations in the number of epochs trained for each fold, ensuring optimal convergence and performance.

## Findings & Problems Version 5:

****

Figure 49: Accuracy & Loss fold 1 Version 5

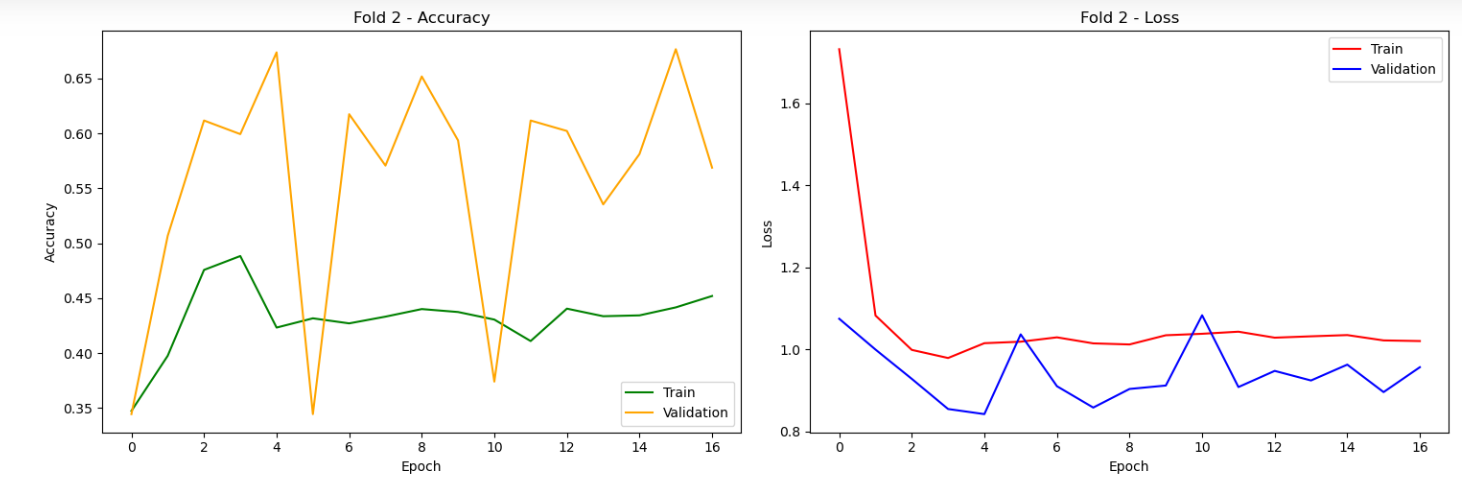
****

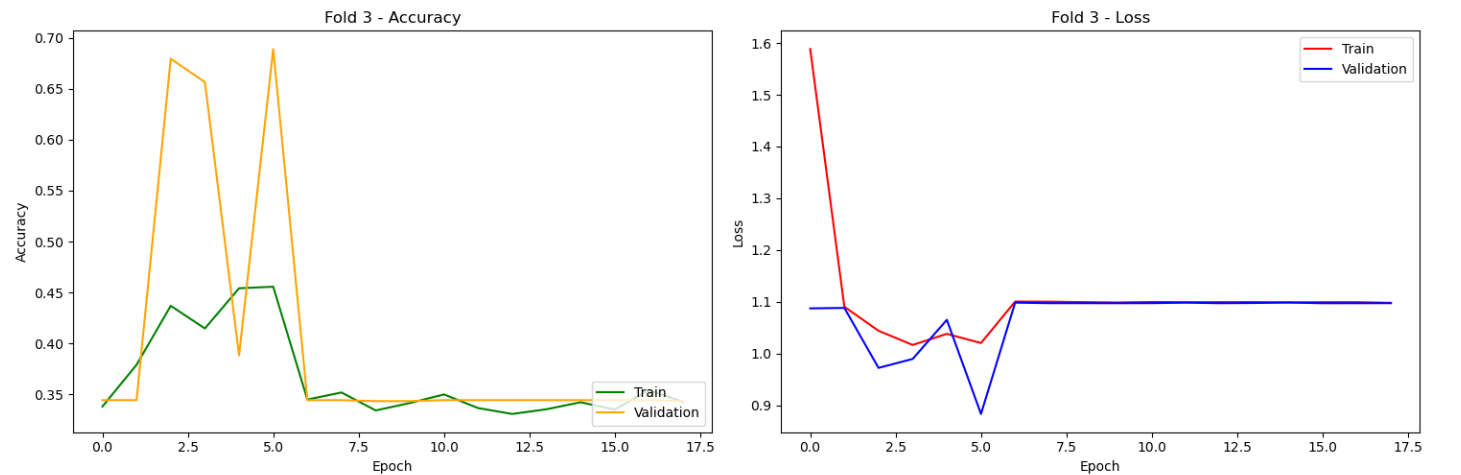
Figure 50: Accuracy & Loss fold 2 Version 5****

Figure 51: Accuracy & Loss fold 3 Version 5

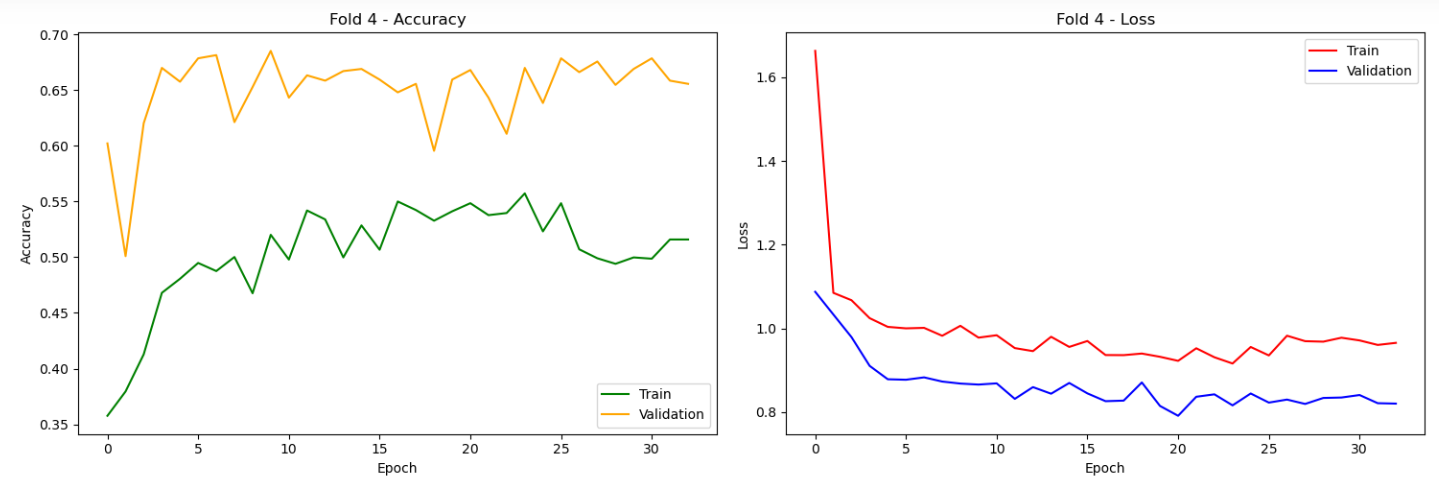
****

Figure 52: Accuracy & Loss fold 4 Version 5

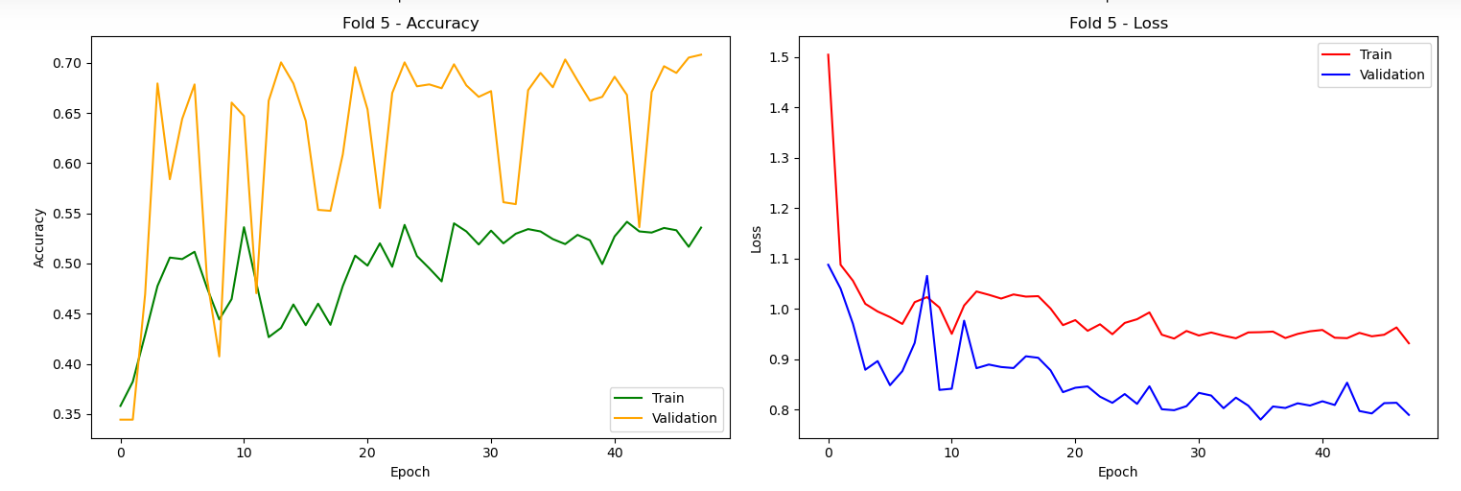
****

Figure 53: Accuracy & Loss fold 5 Version 5

* Utilizing ResNet50 in this instance, applied to a 5-fold cross-validation dataset, yielded a maximum accuracy of 68%.
* However, notable misclassification of pneumonia instances occurred, potentially attributed to inadequate data for training such a large model.
* Despite maintaining consistency in other parameters, further exploration and potential adjustments may be necessary to address these misclassifications and enhance model performance.

## Findings & Problems Version 6:

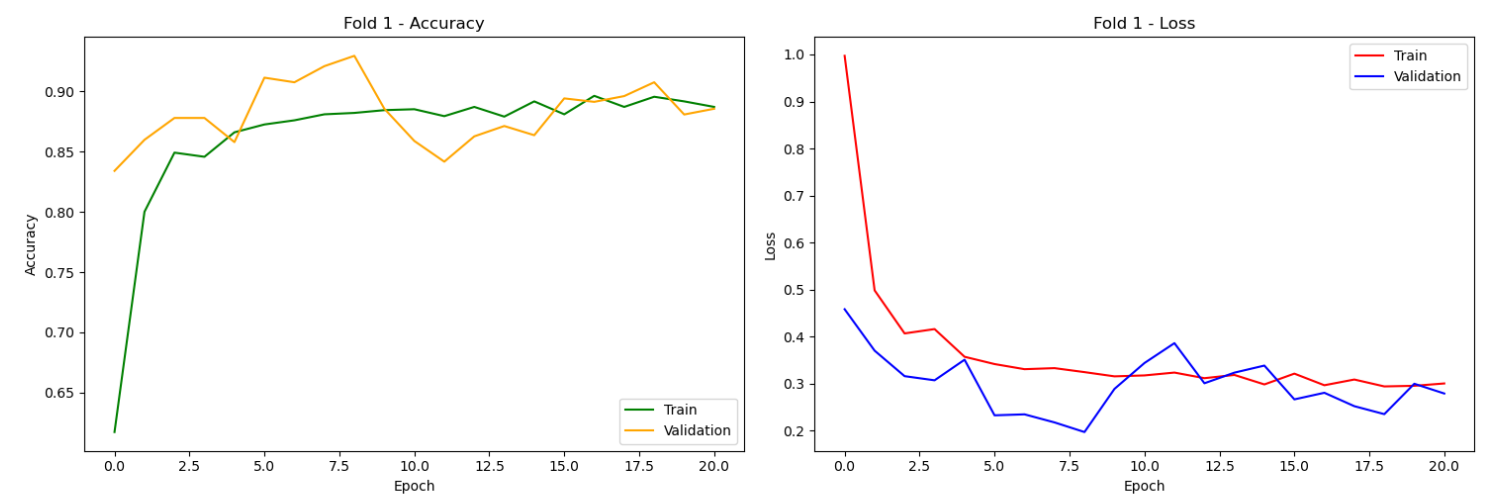
****

Figure 54: Accuracy & Loss for Fold 1 Version 6

****

Figure 55: Accuracy & Loss for Fold 2 Version 6

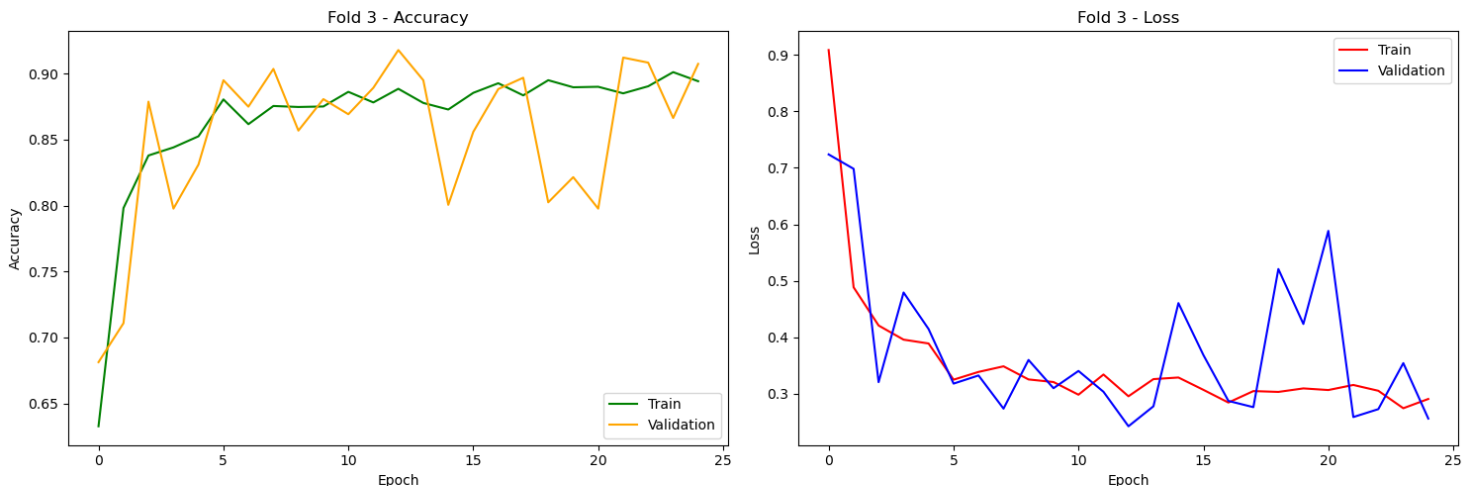
****

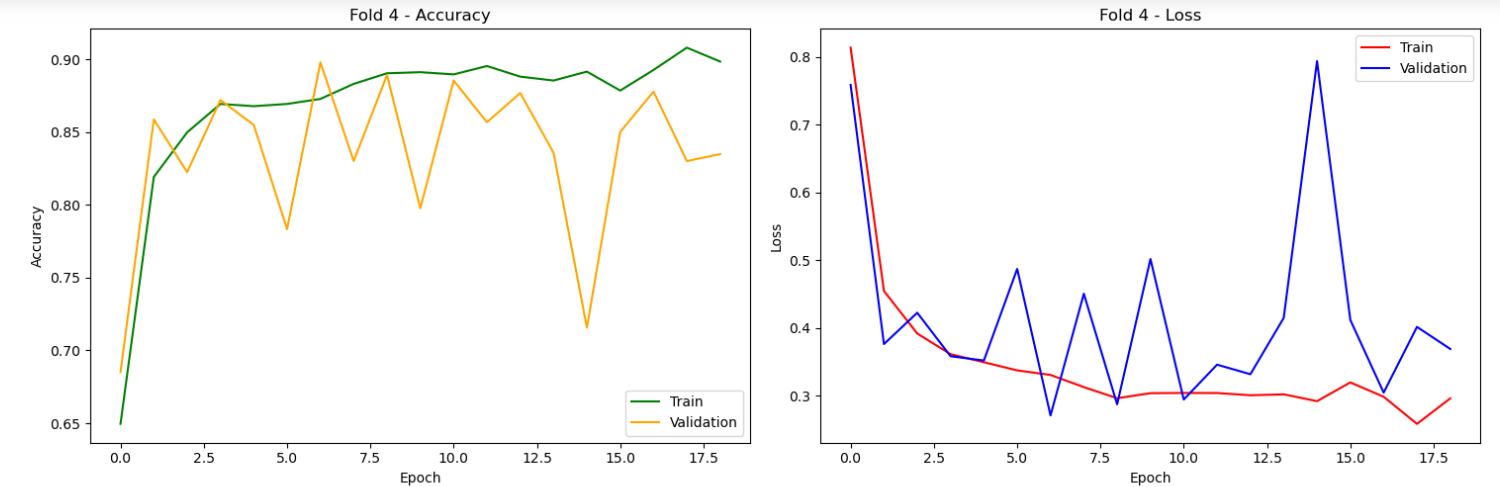
Figure 56: Accuracy & Loss for Fold 3 Version 6****

Figure 57: Accuracy & Loss for Fold 4 Version 6

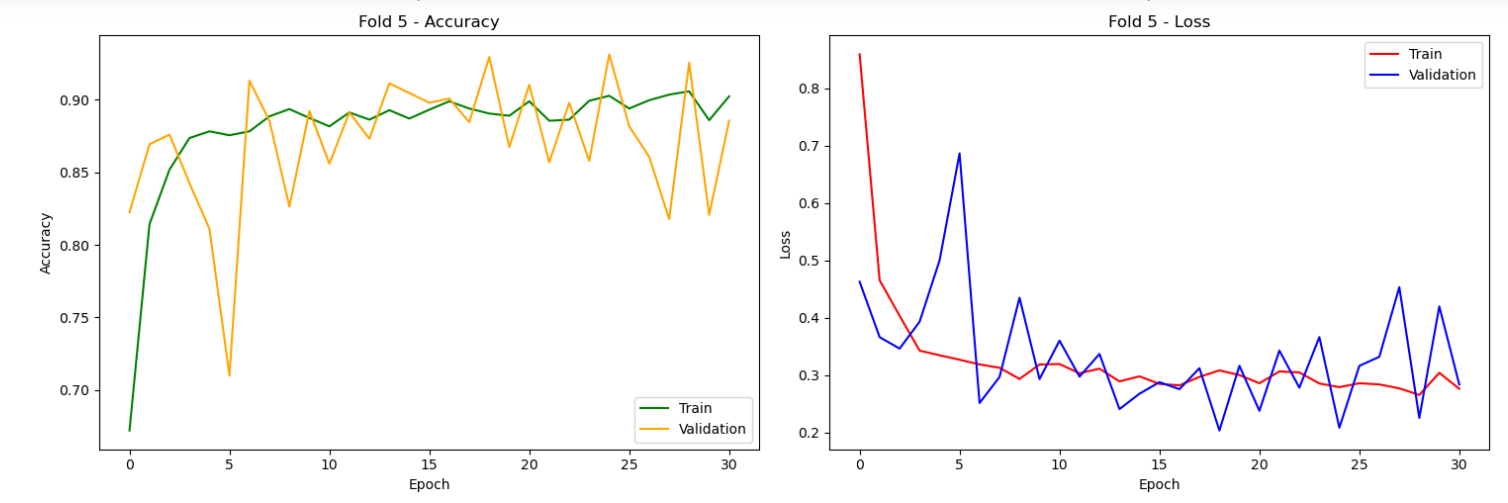
****

Figure 58: Accuracy & Loss for Fold 5 Version 6

* MobileNet version employs a compact architecture with efficient depth-wise separable convolutions.
* Its streamlined design ensures computational efficiency while maintaining high performance.
* During fold 4 of cross-validation, MobileNet achieved its highest accuracy 57% on fold 4, showcasing its effectiveness in resource-constrained environments.

## Findings & Problems Version 7:

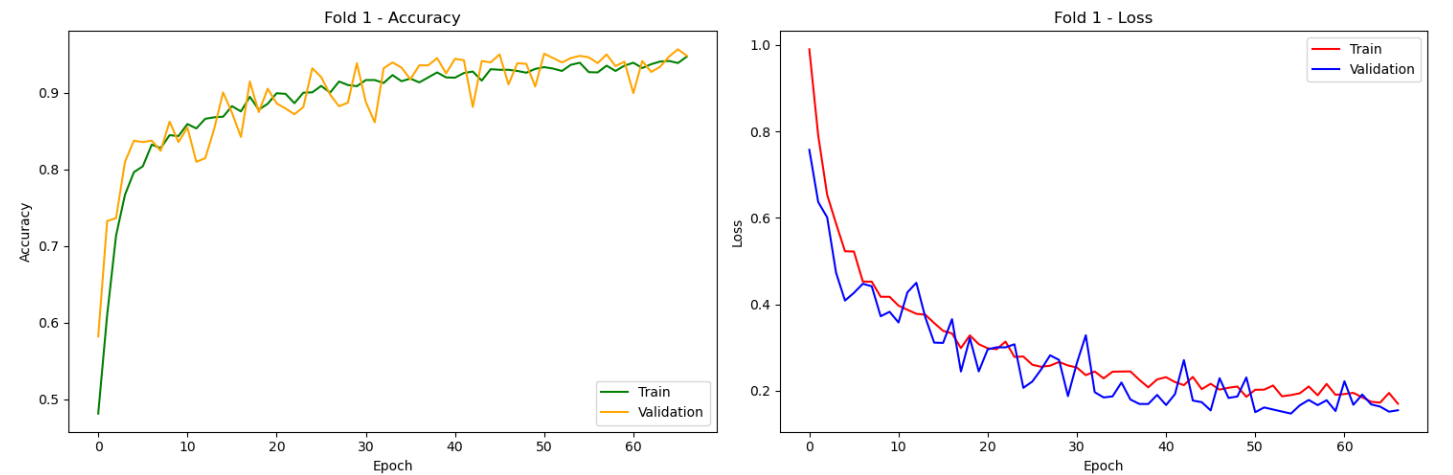
****

Figure 59: Accuracy & Loss fold 1 Version 7

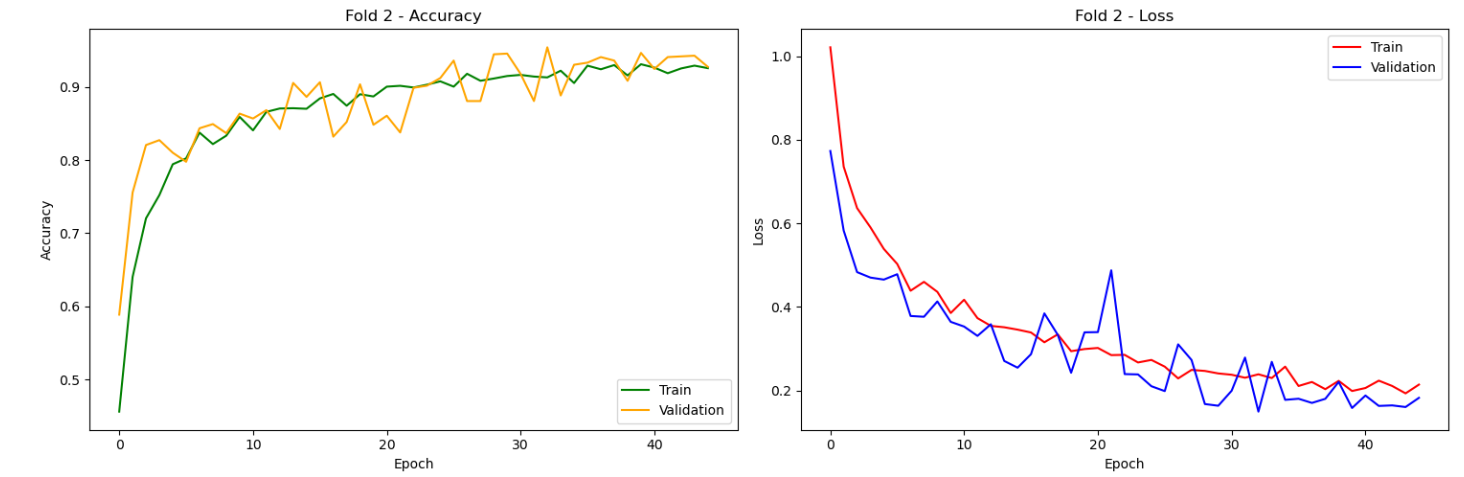
****

Figure 60: Accuracy & Loss fold 2 Version 7

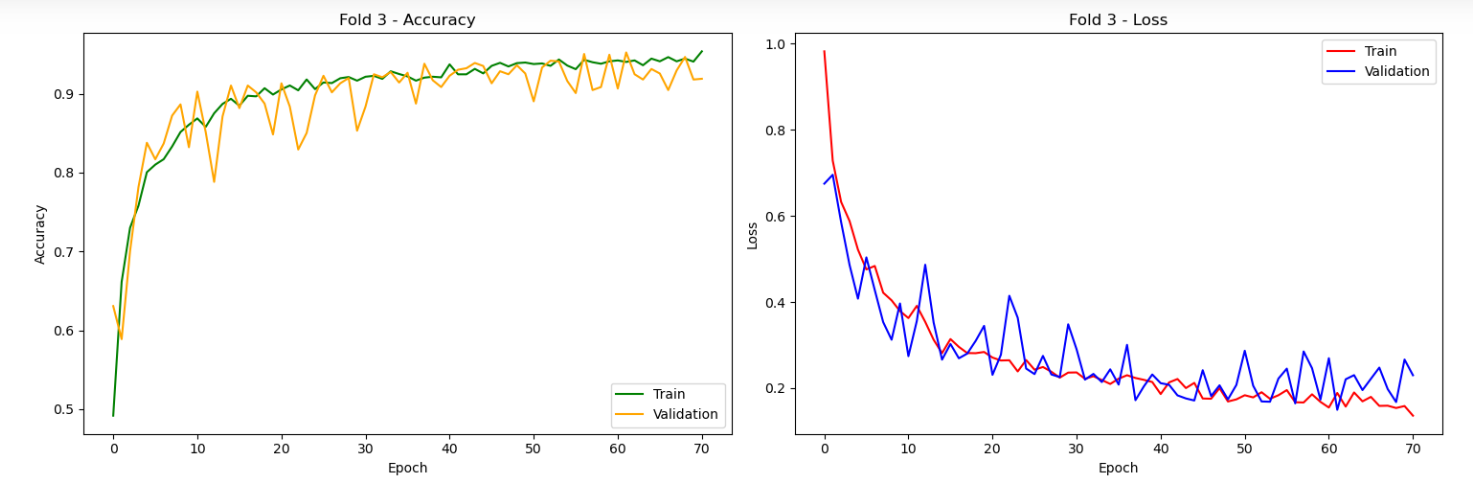
****

Figure 61: Accuracy & Loss fold 3 Version 7

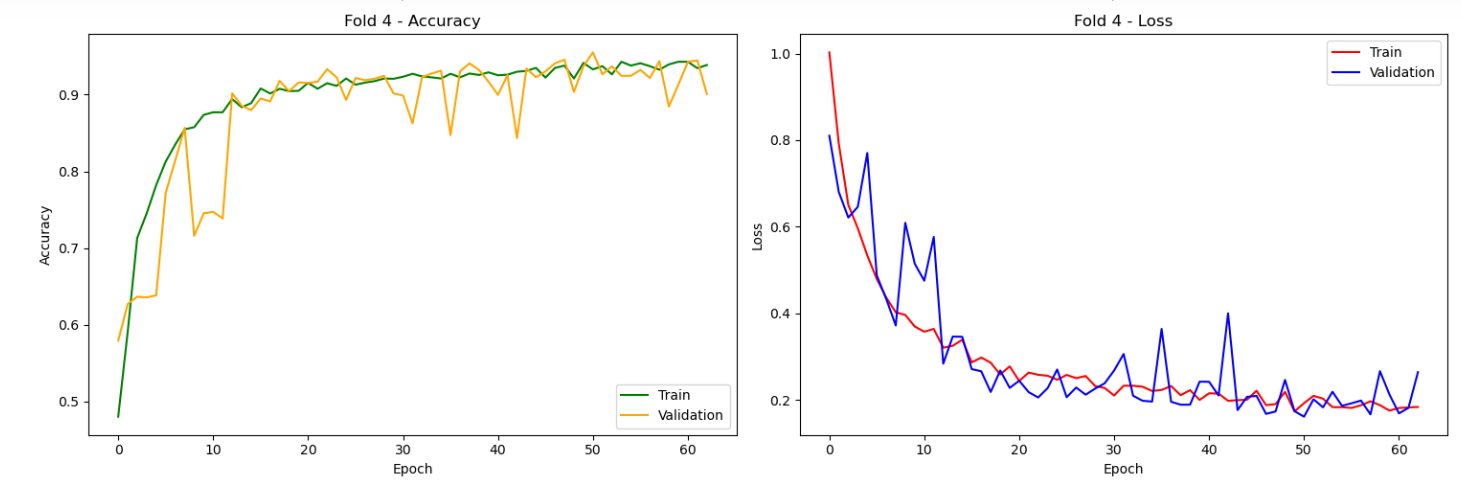
****

Figure 62: Accuracy & Loss fold 4 Version 7

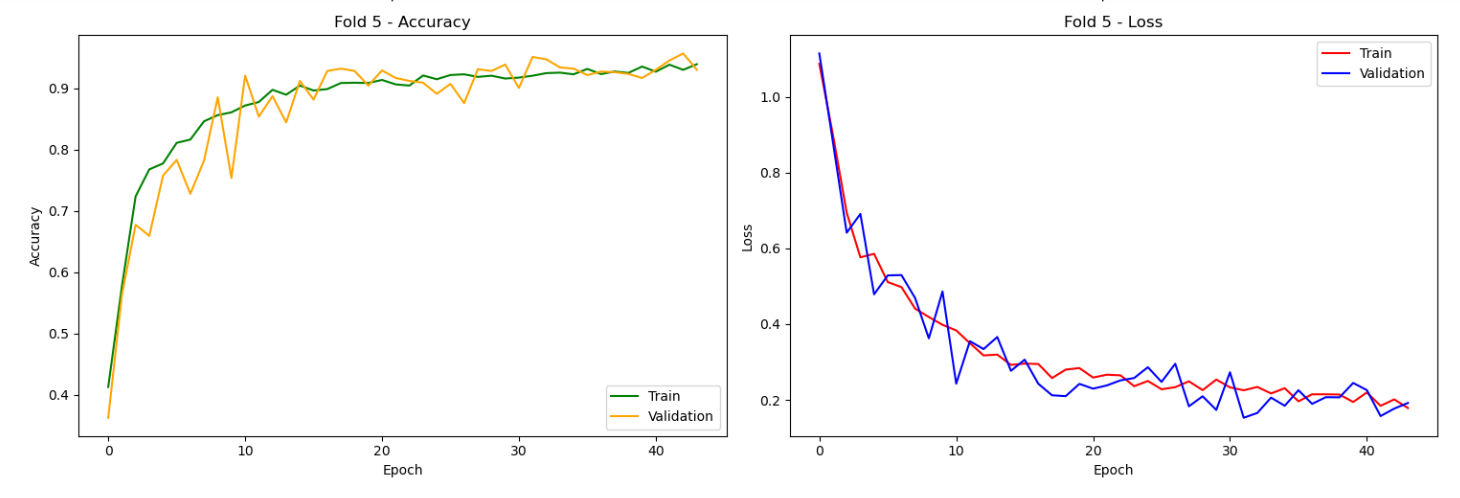
****

Figure 63: Accuracy & Loss fold 5 Version 7

* Our self-defined model architecture comprises 5 Conv2D layers with ReLU activation and MaxPooling2D layers, organized into convolutional blocks, followed by 1 Flatten layer.
* Additionally, the model includes 2 Dense layers with ReLU activation and Dropout layers, culminating in 1 Dense layer with softmax activation serving as the output layer.
* Impressively, the model achieved its highest accuracy of 96% on fold 4 during cross-validation, underscoring its effectiveness

# Project Timeline

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Aug-23 | Sep-23 | Nov-23 | Dec-23 | Jan-23 | Feb-23 | Mar-23 | Apr-23 | May-23 |
| **Abstract** |  |  |  |  |  |  |  |  |  |
| **Proposal** |  |  |  |  |  |  |  |  |  |
| **SRS, SDS** |  |  |  |  |  |  |  |  |  |
| **Data-Preprocessing** |  |  |  |  |  |  |  |  |  |
| **Model Training** |  |  |  |  |  |  |  |  |  |
| **Model Evaluation** |  |  |  |  |  |  |  |  |  |
| **Front-end** |  |  |  |  |  |  |  |  |  |
| **API creation & Back-end** |  |  |  |  |  |  |  |  |  |
| **Integrate Model** |  |  |  |  |  |  |  |  |  |
| **Deployment** |  |  |  |  |  |  |  |  |  |

# Conclusion and Future Work

In the future, there will be opportunities to further enhance the capabilities of our project. One such improvement may be to extend the model's detection range to include more respiratory diseases than just COVID-19 and pneumonia. Furthermore, the incorporation of real-time feedback from radiologists into the classification process presents significant opportunities to improve accuracy and fine-tune diagnostic results. Our project hopes to become a vital resource for both the public and healthcare professionals by pursuing these further improvements. By using our improved approach to detect diseases early, we hope to enhance patient outcomes and support efforts to lessen the effects of COVID-19 and pneumonia. Our goal is to make a significant contribution to the development of healthcare procedures, promoting a more robust and healthy global community.

Our project aimed to develop a user-friendly, interactive AI web-based system capable of accurately diagnosing Pneumonia and COVID-19 from X-ray images. Among the various models evaluated, VGG-16 emerged as the top performer, achieving an impressive accuracy of 95%. This success can be attributed to VGG-16's deep architecture, which excels in extracting intricate features from images, thereby enabling precise classification. Furthermore, the addition of custom Dense layers enhanced the model's capacity to capture subtle patterns and nuances in the X-ray images, contributing to its superior performance compared to other models. By selecting VGG-16 and incorporating additional layers, we were able to surpass our project's objectives by delivering a highly accurate and efficient diagnostic tool for healthcare professionals and patients alike.

Our approach outperformed the previous work mentioned in Literature Review, by addressing several key objectives. Firstly, our comprehensive study of object classification techniques allowed us to identify the most suitable model architecture for our task. Secondly, through empirical evaluation, we determined that VGG-16 with custom Dense layers yielded the best results, surpassing the performance of other models examined. Additionally, our model's ability to accurately classify X-ray images into three categories (Pneumonia, COVID-19, or normal) demonstrates its robustness and reliability. By achieving these objectives, we have developed a state-of-the-art diagnostic system that not only meets but exceeds the requirements mentioned in Problem Identification section, offering significant advancements in the field of medical image analysis and disease diagnosis.

# References

A. Victor Ikechukwu, S. M., R. Deepu, R.C. Shivamurthy. (2021). ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. Retrieved from <https://doi.org/10.1016/j.gltp.2021.08.027>

Adrian Brady, R. Ó. L., Peter McCarthy, and Ronan McDermott. (2012). Discrepancy and Error in Radiology: Concepts, Causes and Consequences. Retrieved from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3609674/>

Daniel Goyal1, M. I.-K., 3, Fatam Mansab4, Amir Iqbal5, Brian McKinstry6, Adeeb P Naasan7, Colin Millar7, Stephen Thomas8, Sohail Bhatti4, Daniel Lasserson9,10 and Derek Burke11. (2021). Improving the early identification of COVID-19 pneumonia: a narrative review

Elene Firmeza Ohata, G. M. B., João Victor Souza das Chagas, Aloísio Vieira Lira Neto, Adriano, Bessa Albuquerque, V. H. C. d. A., Senior Member, IEEE, and, & Pedro Pedrosa Rebouças Filho, M., IEEE. (2021). Automatic Detection of COVID-19 Infection Using

Chest X-Ray Images Through Transfer Learning. Retrieved from <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9205687&tag=1>

Erdi Çallı , E. S., Bram van Ginneken, Kicky G. van Leeuwen, Keelin Murphy. (2021). Deep learning for chest X-ray analysis: A survey. Retrieved from <https://doi.org/10.1016/j.media.2021.102125>

Linda Wang, A. W. (2020). COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. Retrieved from <https://arxiv.org/abs/2003.09871>

Nayana Ambardekar, M. (2022). X-Rays. Retrieved from <https://www.webmd.com/a-to-z-guides/what-is-x-ray>

Organization, W. H. (2023). Deaths in Pakistan due to Covid-19. Retrieved from <https://covid19.who.int/region/emro/country/pk>

S. Cournane a, R. C. b., D. Creagh c, D.G. Byrne b, N. Sheehy d, B. Silke. (2016). Radiology imaging delays as independent predictors of length of hospital stay for emergency medical admissions. *Clinical Radiology.* Retrieved from <https://doi.org/10.1016/j.crad.2016.03.023>

Sachin Kumar, S. S., Shilpa Mahajan, Kuljeet Singh, Surbhi Gupta, Rajneesh Rani, Neeraj Mohan, Vibhakar Mansotra. (2022). LiteCovidNet: A lightweight deep neural network model for detection of COVID-19 using X-ray images. Retrieved from <https://doi.org/10.1002/ima.22770>

Shastri, S., Kansal, I., Kumar, S. et al. (2022). CheXImageNet: a novel architecture for accurate classification of Covid-19 with chest x-ray digital images using deep convolutional neural networks. Retrieved from <https://rdcu.be/dB6Uy>

United Nations International Children's Emergency Fund. (2019). Pneumonia – a preventable disease – kills more children than any other infection. Retrieved from <https://www.unicef.org/pakistan/press-releases/one-child-dies-pneumonia-every-39-seconds-agencies-warn>

Varde, A. S., Karthikeyan, D. & Wang, W. . (2023). Facilitating COVID recognition from X-rays with computer vision models and transfer learning. Retrieved from <https://doi.org/10.1007/s11042-023-15744-9>

World Health Organization. (2020). TOTAL CONFIRMED DEATHS DUE TO COVID-19. Retrieved from <https://ourworldindata.org/grapher/total-daily-covid-deaths>